

IN-32-CR  
204788  
(NAG5-1612)  
177P

## TRANSMISSION ON ATM NETWORKS

by  
Yun-Chung Chen

### A DISSERTATION

Presented to the Faculty of  
the College in the University of Nebraska  
in partial fulfillment of Requirements  
for the Degree of Doctor of Philosophy

Departmental Area of Engineering (Electrical)

Under the Supervision of  
Prof. Khalid Sayood

University of Nebraska-Lincoln

Lincoln, Nebraska  
December 1993

(NASA-CR-195124) VIDEO  
TRANSMISSION ON ATM NETWORKS Ph.D.  
Thesis (Nebraska Univ.) 177 p

N94-29106

Unclass

G3/32 0204788

DISSERTATION TITLE

Video Transmission on ATM Networks

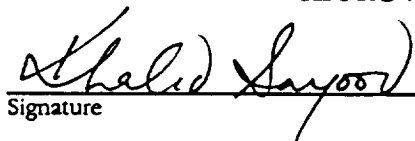
BY

Yun-Chung Chen

SUPERVISORY COMMITTEE:

APPROVED

DATE

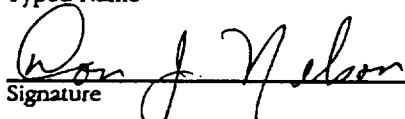


Signature

Khalid Sayood

Typed Name

Aug 31, 1993

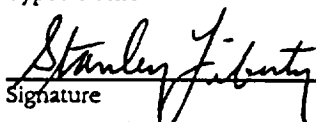


Signature

Don J. Nelson

Typed Name

Aug 31, 1993

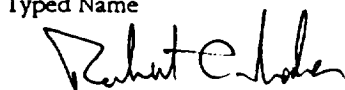


Signature

Stanley Liberty

Typed Name

Aug 31, 1993

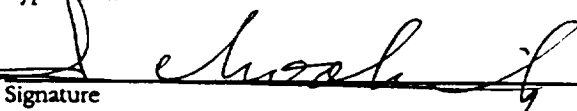


Signature

Robert Maher

Typed Name

Aug 31, 1993



Signature

Fred Choobineh

Typed Name

8/31/93

Signature

Typed Name



GRADUATE COLLEGE  
UNIVERSITY OF NEBRASKA

# VIDEO TRANSMISSION ON ATM NETWORKS

Yun-Chung Chen, Ph.D.

University of Nebraska, 1993

Advisor: Khalid Sayood

The broadband integrated services digital network (B-ISDN) is expected to provide high-speed and flexible multimedia applications. Multimedia includes data, graphics, image, voice, and video. Asynchronous transfer mode (ATM) is the adopted transport techniques for B-ISDN and has the potential for providing a more efficient and integrated environment for multimedia. It is believed that most broadband applications will make heavy use of visual information. The prospect of wide spread use of image and video communication has led to interest in coding algorithms for reducing bandwidth requirements and improving image quality.

This dissertation presents results of our study on the bridging of network transmission performance and video coding. The major results are: 1) Using two representative video sequences, several video source models are developed. The fitness of these models are validated through the use of statistical tests and network queueing performance. 2) A dual leaky bucket algorithm is proposed as an effective network policing function. The concept of the dual leaky bucket algorithm can be applied to a prioritized coding approach to achieve transmission efficiency. 3) A mapping of the performance/control parameters at the network level into equivalent parameters at the video coding level is developed. Based on that, a complete set of principles for the design of video codecs for network transmission is proposed.

## Acknowledgments

I would like to take this opportunity to express my appreciation to Dr. Khalid Sayood for his support and patience. Encouragement from Dr. Don J. Nelson, Dr. Robert Maher, Dr. Stanley Liberty, Dr. Fred Choobineh, and Dr. Gopal Meempat was also appreciated.

Also, I would like to thank the NASA Goddard Space Flight Center (NAG 5-1612) for supporting me during the course of my graduate studies.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>ATM-based B-ISDN</b>	<b>8</b>
2.1	Broadband ISDN . . . . .	9
2.2	Functional characteristics of ATM . . . . .	12
2.3	Classification of B-ISDN services . . . . .	16
2.4	Some Notes . . . . .	18
<b>3</b>	<b>About video coding</b>	<b>19</b>
3.1	Basic image compression techniques . . . . .	21
3.1.1	Pixel coding . . . . .	22
3.1.2	Predictive coding . . . . .	22
3.1.3	Transform coding . . . . .	23
3.1.4	Vector quantization . . . . .	24
3.2	CCITT H.261 video coding standard . . . . .	24
3.2.1	Motivations . . . . .	25
3.2.2	Video coding and multiplexing structure . . . . .	25
3.2.3	Video source coding algorithm . . . . .	28
3.2.3.1	Prediction . . . . .	29

3.2.3.2	Motion compensation . . . . .	29
3.2.3.3	Loop filter . . . . .	30
3.2.3.4	Transformation . . . . .	31
3.2.3.5	Quantization . . . . .	31
3.2.3.6	Zig-zag scanning and run-length coding . . . . .	32
3.2.4	Coding control and rate buffer . . . . .	32
3.2.5	Simulation results . . . . .	33
3.3	Advanced digital television . . . . .	35
3.3.1	Group of pictures . . . . .	36
3.3.2	Input sequencer . . . . .	37
3.3.3	Raster line to block/macroblock converter . . . . .	37
3.3.4	I-frame processing . . . . .	38
3.3.5	P-frame processing . . . . .	38
3.3.6	B-frame processing . . . . .	38
3.3.7	Differential, run-length, and variable-length coding . . . . .	39
3.3.8	Simulation results . . . . .	39
3.4	Subband coding . . . . .	41
3.4.1	Simulation results . . . . .	42
3.5	Mixture block coding with progressive transmission . . . . .	45
3.5.1	Simulation results . . . . .	47
3.6	Some notes . . . . .	50
<b>4</b>	<b>Video source modelling</b>	<b>64</b>
4.1	Video source sequences . . . . .	65
4.2	Models of homogeneous sequence . . . . .	67
4.2.1	Continuous state autoregressive Markov model . . . . .	68

4.2.2	Discrete time Markov chain model . . . . .	69
4.2.3	Discrete state, continuous time birth-death Markov model .	71
4.3	Models of scene-cut sequence . . . . .	76
4.3.1	Another autoregressive and Markov chain model . . . . .	77
4.3.2	Hidden Markov model . . . . .	77
4.4	Goodness-of-fit tests . . . . .	79
4.4.1	Statistics . . . . .	80
4.4.2	Cell loss performance for homogeneous sequence . . . . .	82
4.4.3	Cell loss performance for scene-cut sequence . . . . .	86
4.5	Some notes . . . . .	87
<b>5</b>	<b>Network congestion control</b>	<b>89</b>
5.1	Call admission control . . . . .	91
5.2	Usage parameter control . . . . .	99
5.3	Priority control and selective discard mechanism . . . . .	115
5.4	Explicit congestion notification . . . . .	117
5.5	Traffic shaping . . . . .	118
5.6	Some notes . . . . .	120
<b>6</b>	<b>Video codec design</b>	<b>122</b>
6.1	Call setup . . . . .	122
6.2	Design a traffic shaper . . . . .	127
6.3	Design a rate buffer . . . . .	130
6.4	Packetization . . . . .	132
6.5	Design a priority scheme . . . . .	135
6.6	Adaptive coding based on network status . . . . .	136

6.7	Error control . . . . .	137
6.7.1	Error concealment . . . . .	138
6.7.2	Use of CLP bit . . . . .	142
6.7.3	Partial local decoding . . . . .	142
6.7.4	Other possible approaches . . . . .	147
6.8	Some notes . . . . .	148
7	Conclusions	150
	Bibliography	153



# List of Figures

2.1	ATM cell structure . . . . .	14
2.2	B-ISDN protocol stack at UNI . . . . .	15
3.1	Scanning format of H.261 layer structure . . . . .	27
3.2	Structure of macroblock layer . . . . .	28
3.3	Block diagram of H.261 video codec . . . . .	29
3.4	Example of block matching technique . . . . .	30
3.5	Zig-zag scan . . . . .	32
3.6	An example of a Group of Pictures . . . . .	37
3.7	The structure of a three-dimension subband analysis system . .	42
3.8	An example of quad-tree structure . . . . .	47
3.9	Coding rate of <i>Susie</i> sequence using H.261 algorithm . . . .	51
3.10	PSNR of <i>Susie</i> sequence using H.261 algorithm . . . . .	51
3.11	Coding rate of <i>Football</i> sequence using H.261 algorithm . . .	52
3.12	PSNR of <i>Football</i> sequence using H.261 algorithm . . . . .	52
3.13	Coding rate of <i>Susie</i> sequence using ADTV algorithm . . . .	53
3.14	PSNR of <i>Susie</i> sequence using ADTV algorithm . . . . .	53

3.15	Coding rate of <i>Football</i> sequence using ADTV algorithm . . .	54
3.16	PSNR of <i>Football</i> sequence using ADTV algorithm . . . . .	54
3.17	Coding rate using subband coding algorithm . . . . .	55
3.18	PSNR using subband coding algorithm . . . . .	55
3.19	Coding rate of <i>Susie</i> sequence using MBCPT algorithm . . . .	56
3.20	PSNR of <i>Susie</i> sequence using MBCPT algorithm . . . . .	56
3.21	Coding rate distribution of four passes for <i>Susie</i> sequence . . .	57
3.22	PSNR of four passes for <i>Susie</i> sequence . . . . .	57
3.23	Number of blocks with different coding strategy ( <i>Susie</i> ) . . .	58
3.24	Percentage of coded block for four passes ( <i>Susie</i> ) . . . . .	58
3.25	Coding rate of <i>Football</i> sequence using MBCPT algorithm . . .	59
3.26	PSNR of <i>Football</i> sequence using MBCPT algorithm . . . . .	59
3.27	Coding rate distribution of four passes for <i>Football</i> sequence . .	60
3.28	PSNR of four passes for <i>Football</i> sequence . . . . .	60
3.29	Number of blocks with different coding strategy ( <i>Football</i> ) . .	61
3.30	Percentage of coded block for four passes ( <i>Football</i> ) . . . . .	61
3.31	Susie sequence (original, every tenth frame, left to right, top to bottom) . . . . .	62
3.32	Football sequence (original, every tenth frame, left to right, top to bottom) . . . . .	63
4.1	Coding rate of Sequence 1 . . . . .	66
4.2	Coding rate of Sequence 2 . . . . .	66
4.3	Autocorrelation functions of Sequence 1 and AR(1)-AR(5) . . .	70

4.4	Autocorrelation functions of several models (Sequence 1) . . .	70
4.5	Diagram of state transition rate for birth-death markov model .	71
4.6	Exponential fits ( $e^{-\lambda\tau}$ , $\tau = \text{lag}/30$ ) of autocorrelation function (Sequence 1) . . . . .	72
4.7	Queueing system for minisource model ( $M = 3$ ) . . . . .	74
4.8	Diagram of state transition rate for minisource ( $M = 3$ ) . . . .	74
4.9	Instantaneous transition rate matrix of minisource model . . .	75
4.10	Autocorrelation function of several models (Sequence 2) . . .	78
4.11	Percentile plot of several models (Sequence 1) . . . . .	81
4.12	Percentile plot of several models (Sequence 2) . . . . .	81
4.13	Cell loss probability of several models (Sequence 1, Case 1) . .	83
4.14	Cell loss probability of several models (Sequence 1, Case 2) . .	85
4.15	Cell loss probability of several models (Sequence 2, Case 1) . .	85
5.1	Classification of ATM congestion control mechanisms . . . .	90
5.2	Influence of delay on equivalent bandwidth for one homogeneous sequence . . . . .	97
5.3	Influence of delay on equivalent bandwidth for one scene-cut sequence . . . . .	97
5.4	Equivalent and approximated bandwidth for various number of multiplexed sources (homogeneous sequence) . . . . .	98
5.5	Equivalent and approximated bandwidth for various number of multiplexed sources (sequence with scene-cut) . . . . .	98
5.6	Example of counter state for different mechanisms . . . . .	104

5.7	Influence of counter limit on violation probability using mean bandwidth policing . . . . .	105
5.8	Influence of counter limit on violation probability using equivalent bandwidth policing . . . . .	106
5.9	Overload detection ability of LB, JW, and EWMA mechanisms using equivalent bandwidth policing . . . . .	106
5.10	Percentage of different cells for various overload factors using dual policing scheme . . . . .	109
5.11	Performance of dual policing mechanism under equivalent bandwidth allocation . . . . .	111
5.12	Performance of mean bandwidth policing with marking discipline under equivalent bandwidth allocation . . . . .	111
5.13	Performance of equivalent policing mechanism with discarding discipline under equivalent bandwidth allocation . . . . .	112
5.14	Performance of dual policing mechanism under aggressive bandwidth allocation . . . . .	112
5.15	Performance of mean bandwidth policing with marking discipline under aggressive bandwidth allocation . . . . .	113
5.16	Performance of equivalent policing mechanism with discarding discipline under aggressive bandwidth allocation . . . . .	113
5.17	A prioritized traffic shaping function . . . . .	119
6.1	Cell distribution of 4 passes for Sequence 7 . . . . .	126
6.2	A general video codec . . . . .	128

6.3	A rate buffer with priority mechanism . . . . .	130
6.4	Distribution of high/low priority cells for Sequence 7a . . . . .	131
6.5	Data format of ADTV transport cell . . . . .	132
6.6	Comparison of simulation results w/ and w/o concealment along with number of lost cells (Sequence 3) . . . . .	139
6.7	Concealment and cell loss error (Sequence 3) . . . . .	139
6.8	Comparison of simulation results w/ and w/o concealment along with number of lost cells (Sequence 4) . . . . .	140
6.9	Concealment and cell loss error (Sequence 4) . . . . .	140
6.10	Performance of PSNR vs frame using priority scheme for Sequence 7a . . . . .	143
6.11	Performance of PSNR vs frame using priority scheme for Sequence 8a . . . . .	143
6.12	A codec with partial local decoding . . . . .	144
6.13	Performance of Sequence 7a using priority scheme w/ and w/o PLD . . . . .	145
6.14	Improvement of Sequence 7a using PLD . . . . .	145
6.15	Performance of Sequence 8a using priority scheme w/ and w/o PLD . . . . .	146
6.16	Improvement of Sequence 8a using PLD . . . . .	146
6.17	Error control coding applied perpendicular to the direction of packetization . . . . .	148

# List of Tables

3.1	Video applications and services . . . . .	20
3.2	Performance of coding rate and PSNR using H.261 coding scheme . . . . .	34
3.3	Performance of coding rate and PSNR using ADTV coding scheme . . . . .	40
3.4	Bit rate distribution among subbands for <i>Susie</i> sequence . . .	44
3.5	Performance of coding rate and PSNR using subband coding scheme . . . . .	44
3.6	Performance of coding rate and PSNR using ADTV coding scheme . . . . .	48
3.7	Bit rate distribution among passes ( <i>Susie</i> , MC_on, $T_1=10$ , $T_2=5$ ) . . . . .	49
4.1	Statistics with 95% confidence interval . . . . .	79
4.2	Cell loss probability of several models (Sequence 1, Case 1) . .	82
4.3	Cell loss probability of several models (Sequence 2) . . . . .	86
6.1	Traffic metric and equivalent bandwidth for several video sequences . . . . .	124

# Acronyms and Abbreviations

- **AAL** - *ATM Adaptation Layer*
- **ABR** - *Average Bit Rate*
- **ATM** - *Asynchronous Transfer Mode*
- **AR** - *Auto Regressive*
- **BECN** - *Backward Explicit Congestion Notification*
- **B-ISDN** - *Broadband - Integrated Services Digital Network*
- **CBP** - *Coded Block Pattern*
- **CBR** - *Constant Bit Rate*
- **CCITT** - *International Telegraph and Telephone Consultative Committee*
- **CCIR** - *International Radio Consultative Committee*
- **CIF** - *Common Intermediate Format*
- **CLP** - *Cell Loss Probability*
- **CS** - *Convergence Sublayer*
- **DCT** - *Discrete Cosine Transform*
- **DPCM** - *Differential Pulse Code Modulation*
- **ECN** - *Explicit Congestion Notification*
- **EOB** - *End of Block*

- *EWMA* - *Exponentially Weighted Moving Average*
- *FECN* - *Forward Explicit Congestion Notification*
- *GFC* - *Generic Flow Control*
- *GOP* - *Group of Pictures*
- *GOB* - *Group of Blocks*
- *GQUANT* - *GOB Quantizer Information*
- *HDTV* - *High Definition Television*
- *HEC* - *Header Error Check*
- *HMM* - *Hidden Markov Model*
- *HP* - *High Priority*
- *ISO* - *International Standard Organization*
- *JPEG* - *Joint Photographic Experts Group*
- *JW* - *Jumping Window*
- *LAN* - *Local Area Network*
- *LB* - *Leaky Bucket*
- *LOT* - *Lapped Orthogonal Transform*
- *LP* - *Low Priority*
- *MA* - *Moving Average*
- *MB* - *Macroblock*
- *MBA* - *Macroblock Address*
- *MBC* - *Mixture Block Coding*
- *MBCPT* - *Mixture Block Coding with Progressive Transmission*
- *MC* - *Motion Compensation*
- *MMPP* - *Modulated Markov Poisson Process*



- *MPEG* - *Moving Pictures Experts Group*
- *MQANT* - *Macroblock Quantizer Information*
- *MTYPE* - *Macroblock Type*
- *MVD* - *Motion Vector Data*
- *NTSC* - *National Television Standards Committee*
- *OSI* - *Open Systems Interconnection*
- *PAR* - *Peak to Average Ratio*
- *PBR* - *Peak Bit Rate*
- *PCM* - *Pulse Code Modulation*
- *PCN* - *Personal Communication Network*
- *PDT* - *Prioritized Data Transport*
- *PLD* - *Partial Local Decoding*
- *PSNR* - *Peak Signal to Noise Ratio*
- *PT* - *Payload Type*
- *QAM* - *Quadrature Amplitude Modulation*
- *QCIF* - *Quarter of Common Intermediate Format*
- *QOS* - *Quality of Service*
- *QS* - *Quantization Stepsize*
- *SAR* - *Segmentation And Reassembly*
- *SNR* - *Signal to Noise Ratio*
- *STM* - *Synchronous Transfer Mode*
- *TCOEFF* - *Transform Coefficients*
- *TJW* - *Triggered Jumping Window*
- *UNI* - *User-Network Interface*

- *UPC* - *Usage Parameter Control*
- *VBR* - *Variable Bit Rate*
- *VC* - *Virtual Channel*
- *VCI* - *Virtual Channel Identifier*
- *VLC* - *Variable Length Coding*
- *VLSI* - *Very Large Scale Integration*
- *VP* - *Virtual Path*
- *VPI* - *Virtual Path Identifier*
- *WAN* - *Wide Area Network*

# Chapter 1

## Introduction

Packet-switched networks were originally invented for carrying computer data, since the bursty nature of such information makes it uneconomical to use continuously-connected circuits. In contrast, speech and video signals have for many years been carried over fixed bit-rate circuit-switched connections although they also have bursty information. The conventional approach in circuit-switched connections is to provide a “dedicated path”, thus reserving a peak bandwidth in advance. With a certain amount of bandwidth capacity assigned to a given source, if the output rate of that source is larger than that capacity, quality will be degraded. On the other hand, if the output rate is less than the reserved bandwidth, the excess channel capacity is wasted and channel efficiency is decreased. Lately, the emergence of new network technology and development of data compression techniques have generated discussions between network and coding specialists concerning the potential advantages of variable bit-rate transmissions over such networks.

There has been considerable interest shown in the general statistical multiplexing of digitally encoded speech signals and particular attention has been given to packet-based

systems. A relatively large number of papers have appeared in the literature, addressing topics such as the delays involved, the associated queueing problems, the effects of packet loss, and the regeneration of lost packets.

*Packet video* is, relatively speaking, a more recent field and has attracted a lot of attention. The antecedents of the current research in packet video, however, date back many years. Coding techniques, using partial replenishment, which generate very variable video bit rates were developed at Bell Laboratories in the late 1960s. Conventional channel sharing by several video sources has also been studied. Switching experts at British Telecom Research Laboratories in England made some of the earliest network proposals for variable bit-rate video in the early 1970s. As always, it takes time and the development of technology for ideas to be accepted. Using variable bit-rate, fixed-quality, instead of fixed bit-rate, variable-quality transmission for video is now the trend of the 1990s.

Several coding schemes which support the packet video idea have been developed in the last decade. Verbiest and Pinnoo [1] proposed a DPCM-based system which consists of an intrafield/interframe predictor, a nonlinear quantizer, and a variable length coder. Their codec obtains stable picture quality by switching between three different coding modes: intrafield DPCM, interframe DPCM, and no replenishment. Ghanbari [2] has simulated a two-layer conditional replenishment codec with a first layer based on a hybrid DCT-DPCM scheme and a second layer using DPCM. This scheme generates two type of packets: “guaranteed packets” contain vital information and “enhancement packets” contain “add-on” information. Darragh and Baker [3] presented a subband codec which

attains user-prescribed fidelity by allowing the encoder's compression rate to vary. The codec's design is based on an algorithm that allocates distortion among the subbands to minimize channel entropy. Kishino *et al.* [4] describe a layered coding technique using discrete cosine transform coding, which is suitable for packet loss compensation. Karlsson and Vetterli [5] presented a subband coder using DPCM with a nonuniform quantizer followed by run-length coding for baseband information, and PCM with run-length coding for the remaining bands. Chen *et al.* [6] present a layered packet video coding algorithm based on a progressive transmission scheme. The algorithm provides good compression and can handle significant packet loss with graceful degradation in the reconstruction sequence.

Along with the rapid development of image compression techniques, network technology is also developing at a fast pace. *Broadband-Integrated Service Digital Network (B-ISDN)* represents the most recent development in the continuing evolution of telecommunication systems. The aim of B-ISDN is to provide an all-purpose, flexible, efficient, and cost-effective environment for all the newly emerging services in an integrated fashion. In order to achieve the aggressive goal which B-ISDN aims at, a promising transfer and switching technique called *Asynchronous Transfer Mode (ATM)* has been adopted. It is believed that ATM will play an important role in expanding the network capabilities toward B-ISDN.

The flexibility of ATM networks provides new opportunities for video communication; at the same time, it also presents a lot of new challenges. The main challenge is the efficient use of network resources and mechanisms in order to achieve a satisfactory

quality performance. That means we need a new coding procedure which not only can fully exploit the network capabilities but also has to react dynamically to changing network status. New circumstances also create new types of impairment; instead of dealing with *symbol error*, now we have to overcome the effect of *cell loss*.

It would be ideal to have a real time simulator for both video codec and network. The interactions between these two elements could then be studied extensively. However, it requires a large amount of effort to build a real time simulator. Given that there is still a lot of uncertainty about proposed video coding algorithms and network protocols, building a simulator that would handle all the different scenarios is not feasible. The approach taken in this dissertation is to deal with each key component in packet video separately. By doing that, we wish to obtain an in-depth understanding of the whole problem and come up with the suitable solutions.

The basic concepts and principles behind B-ISDN and ATM are introduced in Chapter 2. It should be noted, however, that many aspects of B-ISDN and ATM are still uncertain or not agreed upon. Basically, we present a general outline of the future transmission network for video and provide some background knowledge for the following chapters. The material is tutorial in nature so readers familiar with the area may wish to skip this chapter.

Image/video coding techniques which are suitable for packet video are the focus of Chapter 3. Although a video coding algorithm for network transmission has not yet been standardized, some common approaches have been adopted in several proposed schemes. Four video coding schemes will be studied in detail, including *CCITT H.261*, *Advanced*

*Digital Television (ADTV)*, *subband coding*, and *mixture block coding with progressive transmission (MBCPT)*. We have made a few modifications to some of the schemes, so even if the reader is familiar with the proposals they should at least skim this chapter. Extensive simulations were performed to explore the capability of these coding schemes. We also define some specific characteristics (e.g. *mean rate*, *peak rate*, and *burstiness*) of coded video data which describes the traffic flow into the networks in this chapter. Finally, the dynamic behavior of the coding scheme is explored.

The successful transmission of variable bit rate video over ATM networks relies on the interaction between the video coding algorithm and the ATM networks. In Chapter 4 we begin our analysis of this interaction. Two major issues of interest are the effect of network parameters such as delay and loss on the video source and the effect of high bit rate video sources on network performance. The detailed characterization of a single video source is an important first step in any effort to study these issues. In Chapter 4, two video sequences with different characteristics are used to represent a video source; one is homogeneous while the other includes scene-cuts. These two sequences represent two extreme cases which video data flow can present to the network. Based on our need, several simple yet accurate models are proposed for these two sequences. The accuracy of these models can guarantee the correctness of the network simulation which is designed for network performance analysis in Chapter 5.

A thorough introduction to resource allocation and congestion control in ATM networks is provided in Chapter 5. These two aspects of networks determine the efficiency of video transmission. The resource allocation algorithm will dictate the cost

and blocking probability of a connection depending on the traffic's characteristic. An efficient resource allocation scheme increases network utilization and therefore decreases the cost of transmission. A promising approach to resource allocation is equivalent bandwidth allocation [46]. This approach not only describes the required bandwidth for different traffic scenarios based on traffic characteristics and quality of service(QoS) requirements but is easy to manage as well. The congestion control algorithm is a major factor in determining the quality of a call. Usage parameter control (UPC) plays a vital role in monitoring traffic flow and thus maintains a well-operated network situation. However, because of the variety of traffic, it is not an easy job to regulate connection to its agreed-upon contract effectively. None of the schemes proposed to date seems able to do the job. The leaky bucket algorithm is relatively effective, however.

Based on the equivalent bandwidth assignment, we propose a *dual leaky bucket mechanism* with the first bucket monitoring the mean bandwidth and the second one monitoring the equivalent bandwidth. With such a design, a misbehaved connection can be easily detected and network congestion can be prevented effectively (if resource allocation is performed appropriately). Also network utilization is effective with a good resource allocation scheme which takes advantage of multiplexing gain. Other congestion control approaches which have effects in video codec design are also investigated. They are priority scheme, reactive ECN scheme, and traffic shaper. Using the video source models developed in Chapter 4, simulations are performed to justify our design.

Based on the understanding of the transmitting channel, a complete set of design principles for video codec is proposed in Chapter 6. Closely following the concept of the



dual leaky bucket mechanism, a prioritized coding scheme is presented and its performance is studied. Also some combined approaches are adopted to smooth the video output flow and thus requested equivalent bandwidth can be reduced. Finally, some error control algorithms are proposed to combat the effect of cell loss which comes from the nature of packet video.

Chapter 7 includes final remarks and the direction of future research.

## Chapter 2

# ATM-Based B-ISDN

In this chapter we provide a tutorial introduction to ATM-based B-ISDN. Readers familiar with this material may wish to skip this chapter. The continuing advances in the technology of high-capacity optical fiber transmission technology and integrated circuit fabrication are giving rise to a number of new communication services. They include high-speed data exchange and retrieval, high-quality interactive videotex and, in particular, videophone, video conference, and distributive TV, all in addition to the conventional voice service. Most of these services show very specific characteristics in terms of bit rate (constant or variable bit rates, CBR/VBR) or required quality of services like information loss, information delay, data flow control, and end-to-end synchronization. The wide range of bit rates and quality of services (QOS) required provided an impetus for CCITT to recommend a new *Broadband Integrated Service Digital Network* (B-ISDN) to accommodate the emerging demand for broadband services. An objective of the network is to be able to accommodate uncertain changes in service mixes, both at the level of the individual interface and the system as a whole. It is assumed that high-capacity and high-

performance fiber-based transmission facilities will be available to support this environment. The transfer mode chosen by the CCITT as the basis of B-ISDN is called the *Asynchronous Transfer Mode* (ATM). ATM is a high-bandwidth, low-delay, packet-like switching and multiplexing technique. In this chapter, we will give an outline of B-ISDN, the function of ATM, and the services which ATM-based B-ISDN can support. The purpose of this exposition is to provide an understanding of the environment for video transmission.

## 2.1 Broadband ISDN

B-ISDN has been conceived as an all-purpose digital network. It will provide integrated access that will support a wide range of applications in a flexible and cost-effective manner. The network capabilities will include support for [7]:

- ***Interactive and distributed services:*** The network will serve as a common carrier of both interactive and distributed services. These services may include voice, video, and data.
- ***Broadband and narrowband rates:*** The network will be based on a fully optical fiber transmission network operating at about 150 Mb/s at the user network interfaces (UNI) and at 600 Mb/s in the network, with the possibility of increasing bandwidth availability in the future.
- ***Bursty and continuous traffic:*** The network will be able to provide guaranteed bandwidth to meet performance requirements of continuous connections. It will

also provide cheaper service with a lower grade to bursty traffic.

- ***Connection-oriented and connectionless services:*** Most communications are best served by connection-oriented services. Call establishment must precede information transfer. The network will also support connectionless communications, including mail and data-oriented communication.
- ***Point-to-point and complex communications:*** Some services require a single point-to-point connection, either unidirectional or bidirectional, between two end points; others may need connections among multiple users. The network will support both kinds of services.

According to Recommendation I.121, the motivations for evolving from narrowband ISDN to broadband ISDN are [8]:

- *The emerging demand for broadband services.*
- *The availability of high speed transmission, switching and signal processing technologies.*
- *The increasing processing power at the user communication side.*
- *The advances in software application processing in the computer and telecommunication industries.*
- *The need to integrate both interactive and distribution services.*
- *The need to integrate both circuit and packet transfer modes.*
- *The need to provide flexibility to accommodate user and operator requirements.*
- *The need to create a set of CCITT recommendations on the broadband aspects of*

*ISDN.*

The basic principles of B-ISDN are:

- *Asynchronous Transfer Mode will be the B-ISDN transfer mode, independent of the transmission technique at the physical layer.*
- *B-ISDN supports:*
  - *Switched, semi-permanent and permanent, point-to-point, and point-to-multipoint connections.*
  - *Demand, reserved and permanent services.*
  - *Circuit and packet mode services of a mono- and/or multimedia type and of a connectionless or connection-oriented nature and in a bidirectional or unidirectional configuration.*
- *B-ISDN architecture is functionally described and therefore independent of technology and implementation.*
- *B-ISDN contains intelligent capabilities for the purpose of providing advanced service features, supporting powerful operation and maintenance tools, network control and management.*
- *The ISDN access reference configuration is the basis of the B-ISDN access reference configuration.*
- *A layered structure approach is used for B-ISDN protocols.*
- *Change in network capabilities or network performance parameters should not degrade the Quality of Service of the existing services.*

- *Existing interfaces and services must be supported in the future B-ISDN.*
- *New network capabilities may be integrated in B-ISDN to accommodate new user requirements or technological progress.*
- *National specific situations may influence B-ISDN implementation.*

The primary triggers for evolving from narrowband ISDN toward B-ISDN include the increasing demand for high bit rate services, especially image and video services, and the development of technology to support these services. The time schedule for the evolution of B-ISDN may depend on the national situations and on best techno-economic compromises.

## **2.2 Functional Characteristics of ATM**

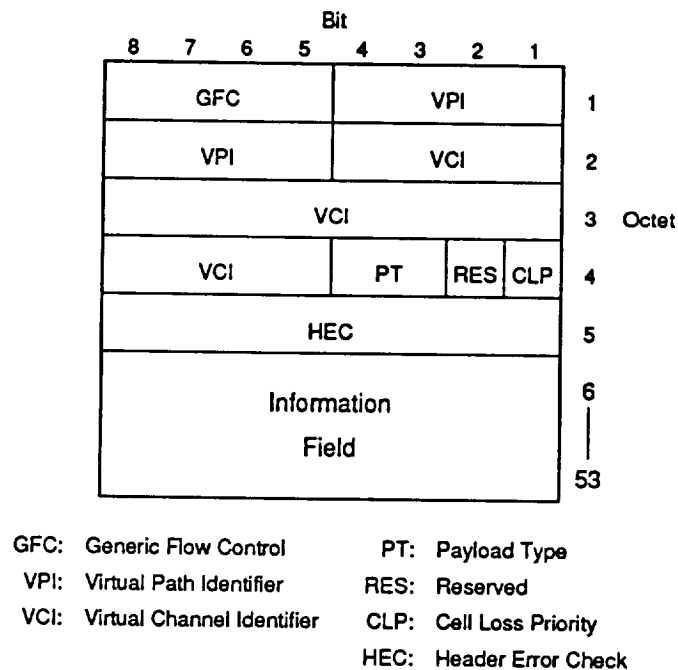
ATM has been adopted by CCITT as the transport technique for B-ISDN and is intended to provide a single common format for transporting voice, video, and data. In the preceding *Synchronous transfer mode* (STM)-based networks, bandwidth is assigned to a service for the duration of a call by allocating time slots within a recurring structure (frame). STM is at its best when it comes to fixed-rate services. However, B-ISDN needs to accommodate various types of service, including bursty transmissions. In ATM, specific periodic time slots are not assigned to a channel. The channel is segmented into fixed-length information-bearing units called cells which can be allocated to services on demand. The cell header identifies which connection the time slot belongs to. Generally ATM is a connection-oriented technique which means that an end-to-end call

establishment procedure is needed. The connection identifiers are attributed at call setup and maintained until the end of the connection. The transfer capacity is assigned on demand at the call setup depending on the source characteristics and on the available resources.

According to Recommendation I.121, adopting ATM as transport technique provides following advantages [8]:

- *High network access flexibility due to the cell transmission principles.*
- *On demand dynamic bandwidth allocation.*
- *Flexible bearer capability allocation and easy provision of semi-permanent connections due to the Virtual Path Concept - unidirectional transport of ATM cells belonging to virtual channels that are associated by a common identifier value.*
- *Independence of the transmission techniques at the physical layer.*

The ATM cell consists of 48 octets of payload and 5 octets of header, as shown in Figure 2.1. The generic flow control (GFC) performs the functions of flow control. The virtual path identifier (VPI) identifies an aggregation of virtual channels. The virtual channel identifier (VCI) is the logical connection identifier. Note that VPI distinguishes the different VP links multiplexed into the same physical layer connection, at a given interface, in a given direction, while VCI identifies a particular VC, in a given VP. Therefore, it is possible for two different VC's belonging to two different VP's at a given interface to have same VCI. In another word, a VC can only be fully identified by both



**Figure 2.1** ATM cell structure.

the VPI and the VCI value. A Virtual Channel/Path Connection (VCC/VPC) is a concatenation of VC/VP links. The payload type (PT) is used to distinguish the user information and network control information. The cell loss priority (CLP) bit is used to indicate the priority class of this cell. The priority indicator is for loss priority rather than delay priority. This means that low priority cells are discarded first when network congestion occurs. The header error check (HEC) is a CRC field to provide error protection for the cell header to minimize misrouting.

The ATM functional architecture comprises two layers [9]:

- **ATM Layer:** *Service and transmission independent layer above the physical layer performing all the functions related to the header field.*



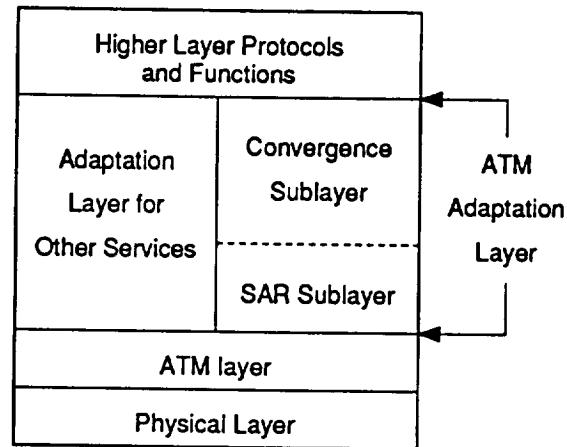


Figure 2.2 B-ISDN protocol stack at UNI.

- **ATM Adaptation Layer (AAL):** *Enhances the services provided by the ATM layer in order to support the functions required by the next higher layer, using AAL-specific information contained in the information field.*

Figure 2.2 shows the protocol stack of B-ISDN at the user network interface (UNI). The ATM layer performs all the functions related with the cell header. The ATM layer functions include [9]:

- **Cell multiplexing and demultiplexing:** *Cell multiplexing consists of the combination of cells from individual VP's and VC's into a non-continuous composite cell flow while cell demultiplexing refers to the inverse process.*
- **Cell header generation/extraction:** *These functions are performed only at ATM layer end-points and consist of the generation of a cell header attached to the information field received from the higher layers, on transmission, and extraction*

*of cell header and delivery of information field to the higher layers, on reception.*

- *Quality of Service provided by the ATM layer. Associated with each VCC/VPC there is one of the Quality of Service classes provided by the network.*

The ATM adaptation layer (AAL) performs the function of mapping user information into ATM cells. There are two sublayers in AAL. The segmentation and reassembly (SAR) sublayer provides functions such as segmentation of information from higher layers into a size suitable for the information field of an ATM cell and reassembly of the ATM cell information field into higher layer information. The convergence sublayer (CS) provides the AAL services at the AAL Service Access Point (SAP). This sublayer is service dependent. Necessary functions for end-to-end synchronization, segmentation of stream-type data, or connection handling for connectionless type services is provided by the AAL so that, at the ATM layer, all functions can be performed fast and effectively by the hardware support.

## **2.3 Classification of B-ISDN Services**

CCITT Recommendation I.211 classifies the services to be provided by B-ISDN into interactive services and distribution services. **Interactive services** are those in which there is a two-way exchange of information (other than control signaling information) between two subscribers or between a subscriber and a service provider. These include conversational services, messaging services, and retrieval services. **Distribution services** are those in which the information transfer is primarily one way, from service provider

to B-ISDN subscriber. These include broadcast services, where the user has no control over the presentation of information, and cyclical services, which allow the user some measure of presentation control. They are listed as follows [10]:

- **Interactive services:** *Bidirectional exchange of information between users or between users and hosts.*
- **Conversational services:** *Bidirectional communication in real-time (no store-and-forward) end-to-end information transfer from user to user or user to host. The information flow may be bidirectional symmetric, bidirectional asymmetric or even unidirectional. (e.g. videotelephony, videoconference, video surveillance)*
- **Messaging services:** *User-to-user communication between individual users via storage units with store-and-forward, mailbox and/or message handling functions. (e.g. video/document mail service)*
- **Retrieval services:** *Retrieval of information stored in information centres and usually provided for public use. This information is retrieved on an individual basis and is sent on user demand. The information timings are under user control. (e.g. broadband retrieval services for movies, high resolution image, audio information, archival information)*
- **Distribution services:** *Unidirectional flow of information from a given point in the network to other (multiple) locations.*
- **Distribution services without user individual presentation control:** *Continuous flow of information from a central source to an unlimited number of authorized receivers connected to the network. The user is not able to control the start or*

*the order of the sequence of information. (e.g. broadcast services for audio and video)*

- *Distribution services with user individual presentation control: Sequence of information entities with cyclical repetition from a central source to a large number of users. The user is able to control the start and the order of the information presentation. (e.g. full channel broadcast videography)*

This classification is from the view of the network and does not take into account the location of the services or the mode of implementation either in the network or in the terminals.

## **2.4 Some Notes**

Despite the promising integration ability of ATM-based B-ISDN, it also creates a lot of new problems which are not easy to solve. Traffic control and resource management are two critical problems that are most relevant to video transmission. In Chapter 5, we will look at these two problems in detail.

## Chapter 3

# About Video Coding

In the previous chapter, we discussed issues related to networks. In this chapter we look at the other major component of packet video, the video compression algorithm. The flexible capabilities provided by ATM based B-ISDN will broaden the range of quality levels and formats available to users. In this scenario, the user may have access to video in the following formats: analog NTSC (National Television Standards Committee), D-I (digital component), D-2 (digital composite), JPEG (Joint Photographic Experts Group, still image compression standard), MPEG (Motion Picture Experts Group, random-access media compression standard), px64 (Telecommunications video standard H.261), and future HDTV (high definition television) standards [11]. The rapid evolution of video compression algorithms has made new video services through the networks possible. Recently, JPEG and MPEG have become industry standards. Both of them are based on the *discrete cosine transform* (DCT), a digital signal processing technique which compresses video at different ratios depending on whether lossless or lossy compression is required [12]. Table 3.1 [11] shows the possible video services and applications on

Type of Service	Flow of Video Information	Purpose	Typical User Location	Subject Material	Required Spatial Resolution	Required Temporal Resolution
Conversational	Two-way	Video-teleconference	Business conference room	Multiple people/graphics	High	Medium
		Videophone (people)	Office desk	Talking head	Low	Medium
		Videophone (graphics)	Office desk	Graphics	High	Low
Entertainment	One-way	Contribution	TV Studio	Varied	Very high	High
		Distribution	Home	Varied	High	High
		VCR	Home	Varied	Medium	High
Instruction/Information	One-way	Training Education Sales Announcement	Home Business	Varied	High	High

Table 3.1 Video applications and services [11].

ATM networks.

The ultimate goal in the design of a video compression scheme is to minimize the bandwidth requirement for the transmission of a specified quality with relatively low complexity. Meanwhile, in order to achieve global integration in ATM networks, CCITT Recommendation I.211 suggests some ATM aspects relevant for video codec design [10]:

- *Cell information transmission concept*
- *QOS parameters*
- *Network based timing information*
- *Constant and Variable Bit Rate services*
- *Independent call and connection control facilities*

In Section 3.1, we will first introduce some basic video coding techniques which can

be applied to some degree in most coding schemes. Following that, four specific schemes, including CCITT H.261, ADTV, subband coding, and MBCPT, which we have simulated will be investigated in detail. The CCITT H.261 algorithm is designed for videophone and videoconference. ADTV is a system proposed for high-resolution full-motion digital television. Subband and MBCPT are two layered coding schemes which have potential applications in any video service. Simulation results for these schemes will be studied and compared. In Chapter 6, the interactions between the network and these coding schemes will be investigated.

### 3.1 Basic Image Compression Techniques

Image compression aims at minimizing the number of bits required to represent an image. Most popular techniques for image coding fall into one of two categories: *predictive coding* techniques and *transform coding* techniques. Both exploit the *redundancy* in the image. Redundancy is also referred as *predictability*, and *smoothness*. In predictive coding the value of the pixel to be coded is predicted based on pixels already visited. The prediction is performed in an identical manner at both transmitter and the receiver. Therefore the predicted value is available to both transmitter and receiver. The prediction error is then coded and sent to the receiver. Transform coding techniques take advantage of the redundancy by packing a large amount of information into a small number of coefficients. The coefficients that do not contain much information can then be thrown away to reduce the data rate without too much distortion.

### 3.1.1 Pixel Coding

Pixel coding, like *pulse code modulation* (PCM), *entropy coding* and *run-length coding* (RLC), processes pixels independently without considering inter-pixel correlation. In PCM the incoming video signal is sampled, quantized, and generally coded by a fixed-length, for example  $B$  bits, binary code. If the quantized pixels are not uniformly distributed, entropy coding encodes a block of  $M$  pixels containing  $MB$  bits with probabilities  $p_i$ ,  $i = 0, 1, \dots, L - 1$ ,  $L = 2^{MB}$ , by  $-\log_2 p_i$  bits, so that the average bit rate is

$$\sum_i p_i (-\log_2 p_i) = H \quad (3.1)$$

$H$  is referred as entropy of the source. *Huffman coding* is the most efficient entropy coding method which gives a variable-length code. RLC codes the length of the runs of 0s of a binary sequence, and is useful whenever large runs of 0s are expected.

### 3.1.2 Predictive Coding

The philosophy underlying predictive coding is to remove mutual redundancy between successive pixels and encode only the new information. One of the simplest predictive coding systems is *differential pulse code modulation* (DPCM) [18]. An example of DPCM can be outlined as follows [19]:

- The prediction  $\hat{y}_{i,j}$  for the  $(i, j)$ th pixel is calculated as

$$\hat{y}_{i,j} = a_1 y_{i-1,j} + a_2 y_{i,j-1} + a_3 y_{i-1,j-1} \quad (3.2)$$

where  $a_1, a_2, a_3$  are prediction coefficients and  $y_{i-1,j}, y_{i,j-1}, y_{i-1,j-1}$  are pixels already



transmitted to a receiver.

- Prediction error  $e_{i,j}$  for the  $(i, j)$ th pixel is

$$e_{i,j} = y_{i,j} - \hat{y}_{i,j} \quad (3.3)$$

- The prediction error is quantized and transmitted.
- An estimate  $\hat{y}_{i,j}$  is calculated for coding of the next pixel by

$$\hat{y}_{i,j} = \hat{y}_{i,j} + \hat{e}_{i,j} \quad (3.4)$$

where  $\hat{e}$  is the quantized error.

For video coding, temporal redundancy can be exploited by using a predictive scheme like motion compensation.

### 3.1.3 Transform Coding

Transform coding is an alternative to predictive coding. In two dimensions, transform coding provides better performance than predictive coding due to two reasons. First, predictive coding is quite sensitive to changes in the statistics of the data. Second, finite-order causal predictors may never achieve compression ability close to transform coding because a finite-order causal representation of a two-dimensional random field may not exist [13]. Moreover, transform coding is visually less objectionable than predictive coding because distortion due to quantization and channel errors is distributed over the entire block. DCT is the most used transform coding scheme because of its superior performance for highly correlated data. The forward and inverse DCT for an 8x8 block are as follows

$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{i=0}^7\sum_{j=0}^7 f(i,j)\cos\left[\frac{(2i+1)u\pi}{16}\right]\cos\left[\frac{(2j+1)v\pi}{16}\right] \quad (3.5)$$

$$f(i,j) = \frac{1}{4}\sum_{u=0}^7\sum_{v=0}^7 C(u)C(v)F(u,v)\cos\left[\frac{(2i+1)u\pi}{16}\right]\cos\left[\frac{(2j+1)v\pi}{16}\right] \quad (3.6)$$

where

$$\begin{aligned} C(u), C(v) &= \frac{1}{\sqrt{2}} && \text{for } u,v = 0 \\ &= 1 && \text{else} \end{aligned} \quad (3.7)$$

$f(i, j)$ : input/output picture element

$F(u,v)$ : DCT coefficient

### 3.1.4 Vector Quantization

Vector quantization represents an extension of conventional scalar quantization. In vector quantization, instead of processing a scalar value, a vector is selected from a finite list of possible vectors to represent an input vector of samples. Each input vector with length  $N$  can be visualized as a point in an  $N$ -dimensional space. The quantizer is defined by a partition of this space into a set of non-overlapping volumes. The output of the optimal quantizer is then the vector identifying the centroid of that volume. Vector quantization can be used in combination with above coding schemes in video coding.

## 3.2 CCITT H.261 Video Coding Standard

The CCITT H.261 video coding standard has been developed for audiovisual services like

videotelephone and videoconference for digital transmission facilities with a capacity which is a multiple of 64 kbits, hence the name px64 where p is in the range 1 to 30.

### 3.2.1 Motivations

The main reasons for CCITT to propose this recommendation are [14]:

- *there is significant customer demand for videophone, videoconference and other audiovisual services.*
- *circuits to meet this demand can be provided by digital transmission using  $B$ ,  $H_0$  rates or their multiples up to the primary rate or  $H_{11}/H_{12}$  rates where  $B$  channel is 64 kbps,  $H_0$  channel is 384 kbps,  $H_{11}$  channel is 1536 kbps, and  $H_{12}$  channel is 1920 kbps.*
- *ISDNs are likely to be available in some countries that provide a switched transmission service at  $B$ ,  $H_0$  or  $H_{11}/H_{12}$  rate.*
- *the existence of different digital hierarchies and different television standards in different parts of the world complicates the problems of specifying coding and transmission standards for international connections.*
- *a number of audiovisual services are likely to appear using basic and primary rate ISDN access and that some means of intercommunication among these terminals should be possible.*

### 3.2.2 Video Coding and Multiplex Structure

The video coding and multiplexing is arranged in a hierarchical structure with four layers.

From top to bottom the layers are:

- **Picture layer**

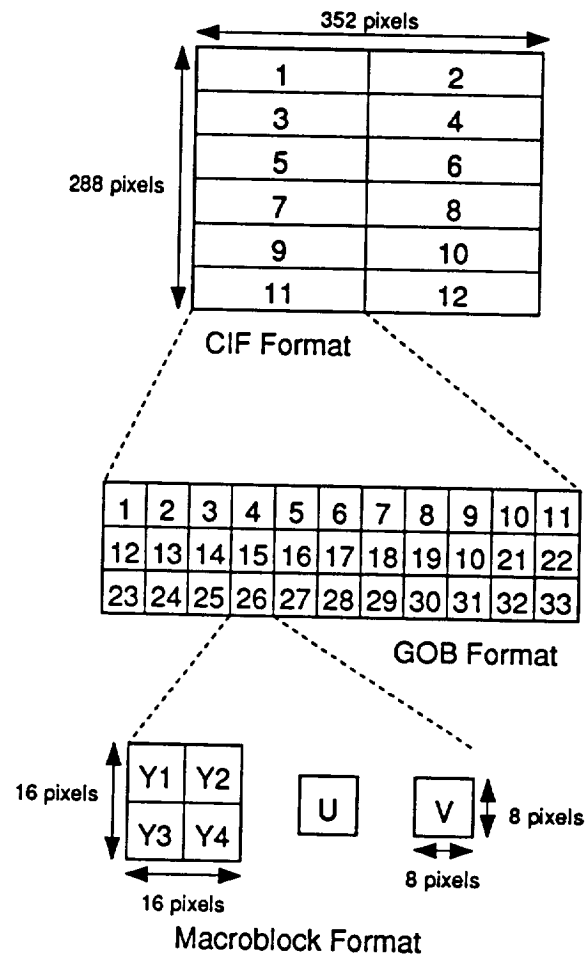
The source coder operates on non-interlaced pictures occurring approximately 29.97 times per second. Pictures are coded as luminance and two colour difference components (Y, U, and V). Two picture scanning formats are specified. In the first format, called Common Intermediate Format (CIF), the luminance sampling structure is 352 pels per line, 288 lines per picture in an orthogonal arrangement. Sampling of each of the two colour difference components is at 176 pels per lines, 144 lines per picture, orthogonal. The picture has an aspect ratio of 4:3. The second format, quarter-CIF (QCIF), has half the number of pels and half the number of lines compared to CIF. All codecs must be able to operate using QCIF. Some codecs can also operate with CIF.

- **Group of Blocks layer (GOB)**

Each picture is divided into group of blocks. A GOB comprises one twelfth of CIF or one third of the QCIF picture areas, as shown in Figure 3.1. A GOB relates to 176 pels by 48 lines of Y and the spatially corresponding 88 pels by 24 lines of each of U and V.

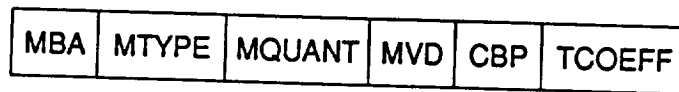
- **Macroblock layer (MB)**

Each GOB is divided into 33 macroblocks, as shown in Figure 3.1. A macroblock consists of 16 pels by 16 lines and the spatially corresponding 8 pels by 8 lines of each of U and V. Figure 3.2 shows the structure of macroblock layer. Data for a macroblock consists of a MB header followed by data for blocks. A variable length



**Figure 3.1** Scanning format of H.261 layer structure.

macroblock address (MBA) indicates the position of a macroblock within a group of blocks. Figure 3.1 also shows the transmission order. For the first transmitted macroblock in a GOB, MBA is the absolute address. For subsequent macroblocks, MBA is the difference between the absolute addresses of the macroblock and the last transmitted macroblock. Macroblocks are not transmitted when they contain no information for that part of the picture. Type information (MTYPE) for each macroblock indicates which data elements are present, including MQANT, MVD,



**Figure 3.2** Structure of macroblock layer.

TCOEFF, and CBP. MQUANT contains information about quantizer. Motion vector data (MVD) is included for all MC macroblocks. MVD is obtained from the macroblock vector by subtracting the vector of the preceding macroblock. The coded block pattern (CBP) gives a pattern number signifying those blocks in the macroblock for which at least one transform coefficient is transmitted. The pattern number is given by:

$$CBP = 32 \cdot P_1 + 16 \cdot P_2 + 8 \cdot P_3 + 4 \cdot P_4 + 2 \cdot P_5 + P_6 \quad (3.8)$$

where  $P_n = 1$  if any coefficient is present for block  $n$ , else 0.

- **Block layer**

A macroblock comprises four luminance blocks and one of each of the two colour difference blocks, as shown in Figure 3.1. Transform coefficients (TCOEFF) data is always present for all six blocks in a macroblock when MTYPE indicates INTRA. In other cases MTYPE and CBP signal which blocks have coefficient data transmitted for them.

### 3.2.3 Video Source Coding Algorithm

The source coder is shown in generalized form in Figure 3.3. The main elements are prediction, block transformation and quantization.

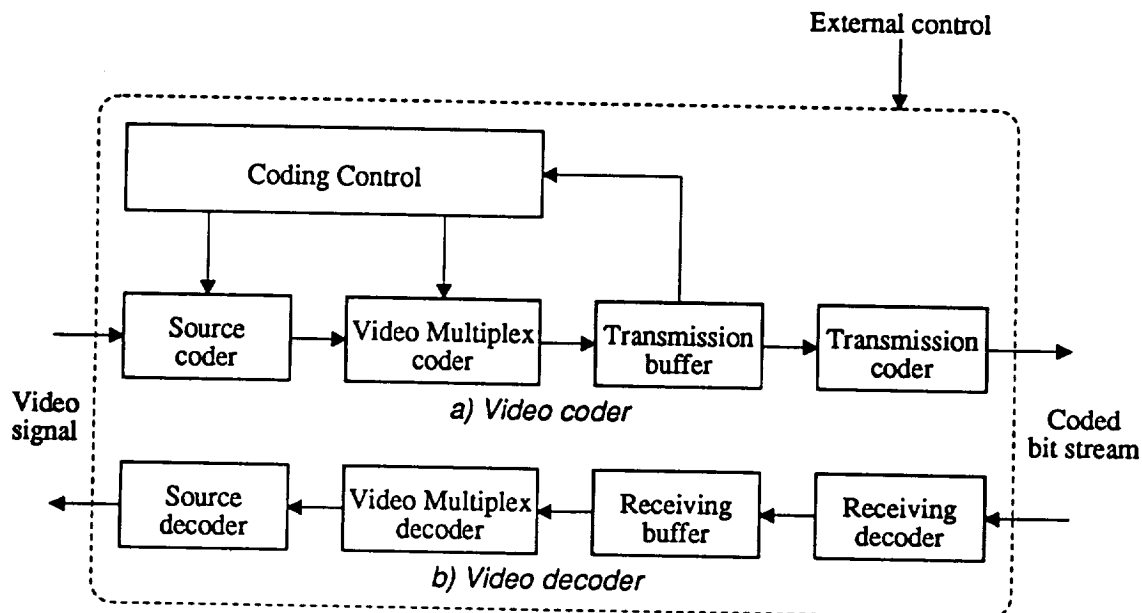
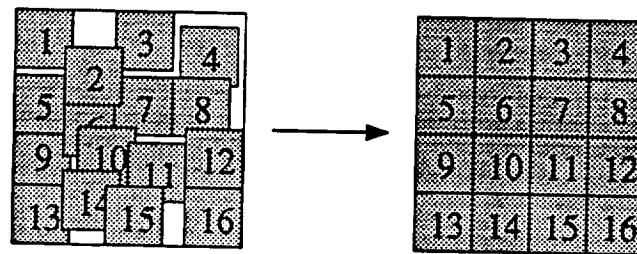


Figure 3.3 Block diagram of H.261 video codec.

### 3.2.3.1 Prediction

Two coding modes, namely INTER and INTRA mode, are suggested by the recommendation. For macroblocks with high temporal correlation, INTER mode coding may be more advantageous. The INTRA mode has been introduced to improve the performance in situations such as scene cuts, fast movements or areas of recovered background. To control the accumulation of inverse transform mismatch and error propagation, some kind of periodic intra coding is forced. The pattern of this forced updating is not defined. The criteria for choice of mode is not subject to recommendation and may be varied dynamically as part of the coding control strategy. The prediction is inter-picture and may be augmented by motion compensation and a spatial filter.

### 3.2.3.2 Motion Compensation



**Figure 3.4** Example of block matching technique.

Motion compensation (MC) is optional in the encoder. The motion is estimated for each macroblock. MC is a simple block matching technique, as shown in Figure 3.4, where a block of pixels in the previous picture which most closely matches the block being encoded is found. A vector, called the motion vector, which describes the displacement of this block relative to the block being encoded may be transmitted. Both horizontal and vertical components of these motion vectors have integer values not exceeding  $\pm 15$ . The vector is used for all four luminance blocks in the macroblock. The motion vector for both colour difference blocks is derived by halving the component values of the macroblock vector and truncating the magnitude to yield integer components. A positive value of the motion vector signifies that the prediction is from pels in the previous picture which are spatially to the right or below the pels being predicted. Motion vectors are restricted such that all pels referenced by them are within the coded picture area.

### 3.2.3.3 Loop Filter

The loop filter is a two-dimensional spatial filter which operates on a predictive (motion-compensated) block. This filter is useful in reducing the high frequency components due



to MC and/or quantization noise in the feedback loop. The filter is separable into two non-recursive one dimensional horizontal and vertical filters both with coefficients of 1/4, 1/2, 1/4, except at block edges where coefficients are changed to 0, 1, 0. The filter is switched on/off for all six blocks in a macroblock according to the MTYPE.

### 3.2.3.4 Transformation

The prediction error or the intra block information is processed before transmission by the separable two-dimensional DCT shown in Eq. (3.5) - Eq. (3.7).

### 3.2.3.5 Quantization

The H.261 algorithm dictates the use of thirty two different quantizers, one for the INTRA dc coefficient and 31 for all other coefficients. Within a macroblock the same quantizer is used for all coefficients except the INTRA dc one. The INTRA dc coefficient is nominally the transform value linearly quantized with a stepsize of 8 and no dead-zone. Each of the other 31 quantizers is also nominally linear but with a central dead-zone around zero and with a stepsize of an even value in the range 2 to 62. For all coefficients other than the INTRA dc one, the reconstruction levels(REC) are in the range -2048 to 2047 and are given by clipping the results of the following formulas:

$$\begin{aligned}
 & \left. \begin{aligned} REC &= QUANT \cdot (2 \cdot level + 1); & level > 0 \\ REC &= QUANT \cdot (2 \cdot level - 1); & level < 0 \end{aligned} \right\} QUANT = \text{"odd"} \\
 & \left. \begin{aligned} REC &= QUANT \cdot (2 \cdot level + 1) - 1; & level > 0 \\ REC &= QUANT \cdot (2 \cdot level - 1) + 1; & level < 0 \end{aligned} \right\} QUANT = \text{"even"} \\
 & REC = 0; \quad level = 0
 \end{aligned} \tag{3.9}$$

QUANT ranges from 1 to 31 and is transmitted by MQUANT.

### 3.2.3.6 Zig-Zag Scan and Run-Length Coding

Since the DCT transform compacts the signal energy on the upper-left corner of the transform coefficient matrix, the quantized transform coefficients are sequentially transmitted following a zig-zag scan shown in Figure 3.5. The coefficients are transmitted as (RUN, LEVEL) pairs, where RUN is the number of

zero coefficients after the previous transmitted coefficient and LEVEL is the quantization level of present non-zero coefficients. All blocks with transmitted coefficients end with the special End of Block (EOB) code.

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

Figure 3.5 Zig-zag scan.

### 3.2.4 Coding Control and Rate Buffer

Several parameters may be varied to control the rate of generation of coded data. These include processing prior to the source coder, the quantizer, block significance criterion and temporal subsampling. The proportions of such measures in the overall control strategy are not subject to recommendation. As a part of coding control, a rate buffer can be used with feedback to control the quantization process. As the buffer fills up, the number of bits assigned to the coefficients is selectively decreased. This technique creates a direct relation between the rate buffer fullness, dependent on the data production, and the quantization step computation by:

$$Quant = 2 \lceil Buffer\ fullness / (200 \cdot p) \rceil + 2 \quad (3.10)$$

where *Quant* is between 2 and 62 and buffer size is  $p \times 6.4$  kbits.

### 3.2.5 Simulation Results

Several simulations have been run to study the coding performance of the H.261 algorithm. The simulation programs were written in *C* and implemented on a *SUN* workstation. The sequences used for testing were the MPEG *Susie* and *Football* sequence. Both sequences contain 150 frames, each of size 240 x 352 pixels (120 x 176 for U, V components) with 8 bits per pixel, which results in a raw bit rate of 30.4 Mbits/s, given a video rate of 30 frames/s. The *Susie* sequence features a woman talking on the phone and contains both low and moderate motion. It represents a videotelephony type of sequence. The *Football* sequence is a TV-like sequence with full motion.

The coding rate and PSNR under different coding conditions are listed in Table 3.2. Two parameters are used to control the coding condition. The parameter  $p$  controls the output rate and length of the rate buffer. The fullness of the rate buffer determines the quantizer stepsize and therefore, the coding rate and quality. The parameter  $p$  has an important impact on both coding rate and quality. The parameter  $T$  is used to decide whether the macroblock after motion compensation needs coding and is calculated as

$$T = \frac{1}{256} \sum_{i=1}^{16} \sum_{j=1}^{16} |x_{i,j} - \hat{x}_{i,j}|$$

(3.11)

$x_{i,j}$ : original pixel value in macroblock  
 $\hat{x}_{i,j}$ : predicted pixel value after motion compensation

From Figures 3.9 - 3.12, we can see that the H.261 algorithm produces a smooth output

	Mean Rate (kbits/sec)	STDR	PAR	Average PSNR	STDP	Subjective Test
Sequence 1	951	29	1.31	38.93	0.72	very good except blocky first frame
Sequence 2	950	30	1.31	38.35	0.49	blocky first frame, blurred in neck, MC artifacts
Sequence 3	574	37	1.65	36.68	0.73	not as sharp as Seq. 1, but O.K., blocky first frame
Sequence 4	951	30	1.31	35.92	0.80	blurry, but O.K. blocky first frame
Sequence 5	952	36	1.40	27.59	0.60	slightly patchy, blocky in grass, acceptable
Sequence 6	952	38	1.42	27.57	0.59	slightly patchy, blocky in grass, blurry
Sequence 7	1912	50	1.20	31.17	0.56	very good
Sequence 8	953	40	1.42	25.15	1.36	extremely grainy, patchy unacceptable

STDR: Standard deviation of coding rate

PAR: Peak to average ratio

STDP: Standard deviation of PSNR

Sequence 1: "Susie", MC\_on, p=15, T=1

Sequence 2: "Susie", MC\_on, p=15, T=3

Sequence 3: "Susie", MC\_on, p=9, T=1

Sequence 4: "Susie", MC\_off, p=15, T=1

Sequence 5: "Football", MC\_on, p=15, T=1

Sequence 6: "Football", MC\_on, p=15, T=3

Sequence 7: "Football", MC\_on, p=30, T=1

Sequence 8: "Football", MC\_off, p=15, T=1

p: Bandwidth = px64 kbits/sec

T: Coding threshold after motion compensation (mean error)

**Table 3.2** Performance of coding rate and PSNR using H.261 coding scheme.

rate with a somewhat bursty PSNR performance. The PSNR is defined as

$$PSNR = 10 \cdot \log \frac{255^2}{MSE} (dB) \quad (3.12)$$

where MSE is the mean squared error. Note that PSNR value is calculated only with the luminance field. From the recorded sequence we did not observe serious chromatic artifacts. Motion compensation improves the performances of both sequences by about 2-3 dB. Increasing the value of  $T$ , as in Sequence 2, creates annoying effects in the smooth background of the *Susie* sequence. However, large  $T$  does not cause visible problems for the full-motion *Football* sequence. Also, blocking effects are observed in the lower regions of first frame of all *Susie* sequences. This is because the first frame is intra-mode coded which leads to the buffer getting filled up as the lower portions of this frame are being coded. Consequently, this means that when coding the lower regions of this frame, the quantizer is coarse. However, the blocking artifacts quickly fade away due to motion compensation and finer quantization in the following frames. Despite the low PSNR values for *Football* sequence, the subjective test is surprisingly good compared with the high PSNR *Susie* sequence since it is relatively difficult to observe coding artifacts due to the fast moving objects in the sequence.

### 3.3 Advanced Digital Television

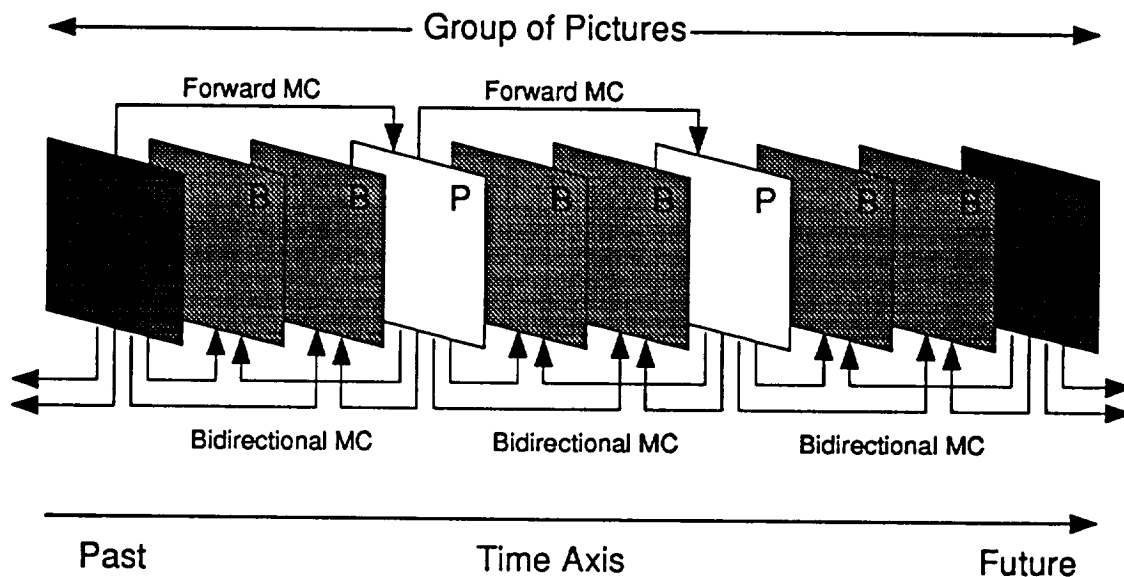
Advanced Digital Television (ADTV) is a proposed high-definition television (HDTV) system. There are three key elements in the ADTV system [15].

- ADTV adopts MPEG++ draft proposal as its compression scheme.
- ADTV incorporates Prioritized Data Transport (PDT) which is a cell relay-based data transport layer to support the prioritized delivery of video data. PDT also offers service flexibility and compatibility to B-ISDN.
- ADTV applies spectral-shaping techniques to Quadrature Amplitude Modulation (QAM) to minimize interference from and to any co-channel NTSC signals.

The basic compression approach of ADTV is the MPEG++ algorithm which upgrades the standard MPEG approach [20] to HDTV performance level. The key components of this algorithm are described below.

### **3.3.1 Group of Pictures (GOP)**

A GOP comprises up to three types of frames, the I, P, and B frames. The I frames are processed using only the intra-frame DCT coder with adaptive quantization; the P frames are processed using a hybrid temporal predictive DCT coder with adaptive quantization and forward motion compensation; the B frames are processed using a hybrid temporal predictive DCT coder with adaptive quantization and bidirectional motion compensation. The I and P frames are referred as the anchor frames because of their roles in the bidirectional motion compensation of the B frames. The GOP structure, as shown in Figure 3.6, offers a good tradeoff between the high efficiency of temporal predictive coding, good error-concealment features of periodic intra-only processing, and fast picture acquisition.



**Figure 3.6** An example of Group of Pictures.

### 3.3.2 Input Sequencer

The GOP data structure requires some unique sequencing of the input video frames. Because of the backward motion compensation in B frames processing, the anchor frames must be processed before the B frames associated with the two anchors. The frames are transmitted in the same order as they are processed.

### 3.3.3 Raster Line to Block/Macroblock Converter

The basic DCT transform unit is an 8x8 pixel block called a block. The basic quantization unit is four adjacent blocks of Y, and one U and one V blocks. Such a quantization unit is called a macroblock, and is similar to the macroblock in the H.261 scheme. The converter converts the raster line format to the block and the macroblock format.

### **3.3.4 I-frame Processing**

An I frame is processed by an intra-frame DCT coder without motion compensation. A fixed quantizer is applied to the DC coefficient. The AC values are first weighted by a down-loadable quantization matrix before uniform adaptive quantization. The quantization step for the AC coefficients is controlled by a rate controller. The I frame coding is pretty much the same as JPEG scheme.

### **3.3.5 P-frame Processing**

A P frame is first processed by a forward motion compensation, motion is always referenced to the nearest past anchor frame. The search area is proportional to the number of B frames between two consecutive anchor frames. Then the prediction residue or original macroblock, depending on the motion compensation result, goes to the DCT coder and quantizer. For intra-macroblocks, the DCT coefficient quantization is identical to that used for the I frames. For motion-compensated macroblocks, the DC and AC coefficients are quantized with same uniform quantizer.

### **3.3.6 B-frame Processing**

Unlike the P frames, the B frames are subjected to bidirectional motion compensation. The motion references are the two anchor frames sandwiching the B frames. The search regions are proportional to the temporal distance between the B frame and the two anchor frames. Like the P-frame macroblocks, the B-frame macroblocks have a number of modes. In addition to all the modes for the P-frame macroblocks, the B-frame



macroblocks further include a bidirectional interpolative mode, using both forward and backward motion compensation, and a unidirectional mode. In the interpolative mode, an average of the forward and the backward motion-compensated macroblocks is used as the prediction macroblock. The B-frame macroblock is processed in the same manner as a P-frame macroblock.

### 3.3.7 Differential, Run-Length, and Variable-Length Coding

The quantized DC coefficients of all the I-frame macroblocks and P-, B-frame macroblocks in intra mode are coded with a DPCM coder. The quantized AC coefficients are coded with run-length coding after the zig-zag scan ordering. Motion vectors are differentially coded. In addition, VLC is applied to all the coded information: motion vectors, macroblock addresses, block types, etc..

### 3.3.8 Simulation Results

The ADTV system described above without the priority and transport processors was simulated in detail. In our ADTV simulator, the frames were arranged in the following sequence

I B B P B B P B B P B B I B B P ...

The coding rate and PSNR are listed in Table 3.3.  $C$  is the parameter used to control the long term coding rate, and plays the same role as  $p$  in H.261 algorithm. The parameter  $T$  is again used to decide whether the macroblock after motion compensation needs

	Mean Rate (Mbits/sec)	STDR	PAR	Average PSNR	STDP	Subjective Test
Sequence 1	0.95	0.087	1.31	37.84	1.17	blurred, annoying blocks in neck
Sequence 2	1.91	0.068	1.12	41.25	0.94	very good
Sequence 3	1.04	0.612	3.24	38.62	1.72	slight grain, better than Seq. 1 overall
Sequence 4	1.91	0.160	1.20	30.20	0.91	blurred, slightly blocky, acceptable
Sequence 5	3.83	0.171	1.12	34.89	1.09	excellent
Sequence 6	2.38	2.080	4.23	31.31	3.74	slight grain, no visible artifacts

Sequence 1: "Susie", C=0.96, T=1

Sequence 2: "Susie", C=1.92, T=1

Sequence 3: "Susie", C=0.96, T=1, QS=4

Sequence 4: "Football", C=1.92, T=1

Sequence 5: "Football", C=3.84, T=1

Sequence 6: "Football", C=1.92, T=1, QS=4

C: Channel Bandwidth, Mbits/sec

T: Coding threshold after motion compensation (mean error)

QS: Quantization step size for I frame

**Table 3.3** Performance of coding rate and PSNR using ADTV coding scheme.

coding. In Sequence 1, with  $C = 0.96$ , the average rate is 0.37 bits/pixel. It is observed that this rate is not sufficient to effectively code the I frames. As the B and P frames depend heavily on the I frames, poorly coded I frames create annoying blocking artifacts which propagates down the entire sequence. When the parameter  $C$  is increased to 1.92 the blocking artifacts are removed as seen in Sequence 2. Due to the importance of the I frame, which serves as the anchor frame for both P and B frame, it may be reasonable to put more coding effort into the I frames to try to eliminate the blocking effect. In

Sequence 3, the ADTV algorithm has been modified to keep the quantization stepsize  $QS$  constant while coding the I frames. One effect is that the buffer becomes really full during coding the I frame, and the subsequent frame gets very little of the coding resources. This results in an increase in burstiness as can be seen from Figure 3.13. However, this approach does result in the reduction/elimination of the blocking effect. From subjective test, Sequence 3 is perceptually more appealing than Sequence 1. The same arguments apply to the Sequences 4 - 6 for *Football* sequence.

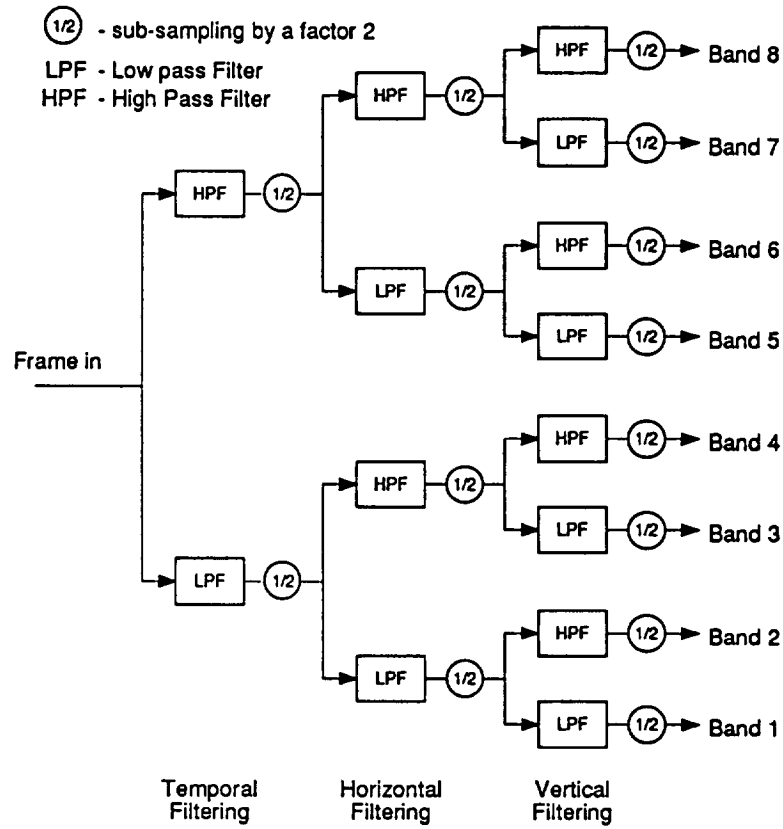
### 3.4 Subband Coding

In subband coding, a signal is passed through a bank of bandpass filters, the analysis filters. Owing to the reduced bandwidth, each resulting component may be subsampled to its new *Nyquist frequency*. Following that, each subband would be encoded, transmitted, and, at the destination, decoded. To finally reconstruct the signal, each subband is up-sampled to the sampling rate of the input. All up-sampled components are passed through the synthesis filter and are added to form the reconstructed signal. Perfect recovery of the original signal is possible if the filters meet certain conditions. The transfer functions of the filters used in our work are as follows [16]

*Temporal filters:*

$$H_l(z) = \frac{1}{2} (1 + z^{-1}), \quad H_h(z) = \frac{1}{2} (1 - z^{-1}). \quad (3.13)$$

*Spatial analysis and synthesis filters:*



**Figure 3.7** The structure of a three-dimension subband analysis system.

$$\begin{aligned}
 H_l(z) &= \frac{1}{4} (-1 + 2z^{-1} + 6z^{-2} + 2z^{-3} - z^{-4}), \\
 H_h(z) &= \frac{1}{4} (1 - 2z^{-1} + z^{-2}), \\
 G_l(z) &= \frac{1}{4} (1 + 2z^{-1} + z^{-2}), \\
 G_h(z) &= \frac{1}{4} (1 + 2z^{-1} - 6z^{-2} + 2z^{-3} + z^{-4}).
 \end{aligned} \tag{3.14}$$

where subscripts  $l$  and  $h$  stand for low pass and high pass filter. For above given filters, a delay of 3 pixels is introduced.

### 3.4.1 Simulation Results

The subband analysis has not resulted in any compression, the sum of data in the subbands equals that of the input. However, it has yielded a desirable separation of the data. In our simulator, eight bands separation is achieved by using the analysis system shown in Figure 3.7. It is noted that the subsampling is performed in the same direction as the filtering operation that it follows. The filtering operations yield one band (band 1 which is through all low pass filters) with an intensity distribution similar to that of the input. The other bands have distributions highly concentrated at or around zero and variance highly reduced compared with that of the input distribution. Note that band 1 has an expanded range of intensity values compared with the 256 permissible levels of the input. This is owing to the gain factor of the spatial analysis low pass filter. Besides, considering each pixel in base band will be up-sampled and interpolated to a 4x4 block of pixels in each of the two frames, care should be taken during quantization to avoid artifacts. The first band is transform coded using DCT and a uniform quantizer without a dead zone. The quantization stepsize is 16 and 32 for luminance and color difference components respectively.

Since the data in the higher bands shows typical Laplacian probability distribution with greatly reduced variance, threshold coding [17] is used to do the compression. The data in different subbands have different variances which determine the importance of that band and how much coding effort we would like to put in. The current implementation has symmetric, uniform quantization for bands 2 to 8. For each band, the width of the dead zone and the quantization stepsize are adjusted. After quantization, run-length coding is applied to all bands. Table 3.4 shows the bit rate distribution among subbands for *Susie*

Band	Mean Rate	Standard Deviation	Maximum Rate	Minimum Rate
1	496.44	87.05	622.38	255.93
2	69.44	15.36	97.05	31.86
3	63.93	32.16	140.22	0.12
4	18.47	8.84	36.84	0.06
5	465.97	217.04	1153.17	162.54
6	32.58	10.50	63.99	17.58
7	38.41	15.10	70.14	0.06
8	10.84	8.46	22.74	0.06
Total	1196.16	117.04	1568.46	983.97

unit: Kbits/sec

**Table 3.4** Bit rate distribution among subbands for *Susie* sequence.

	Mean Rate (Mbits/sec)	STDR	PAR	Average PSNR	STDP	Subjective Test
"Susie"	1.19	0.11	1.31	34.45	1.02	blurred, slightly blocky in neck region
"Football"	4.43	0.45	1.18	28.46	0.27	slightly blurred, no visible artifacts

**Table 3.5** Performance of coding rate and PSNR using subband coding scheme.

sequence. The maximum and minimum rates are the instantaneous rates, which correspond to the number of bits needed to encode a particular frame in the sequence. It is noticed that band 1 and 5 consume most of the coding resources. Band 5 also shows a much more bursty output than band 1. Table 3.5 shows the overall performance for both sequences. Both sequences are somewhat blurred because of less contribution from the higher bands. Frame by frame performance is demonstrated in Figures 3.17 and 3.18.

### 3.5 Mixture Block Coding with Progressive Transmission

Mixture Block Coding (MBC) is a variable-blocksize transform coding algorithm which codes the image with different blocksizes depending upon the complexity of that block area. Low-complexity areas are coded with a large blocksize transform coder while high-complexity regions are coded with small blocksize. The complexity of a specific block is determined by the distortion between the coded and original image. The advantage of using MBC is that, depending on the complexity or business of the region being coded, MBC has the ability to choose a finer or coarser coding scheme for different parts of the same image. With the same rate, MBC is able to provide an image of higher quality than a coding scheme which codes regions of varying complexity with the same blocksize.

When using MBC, the image is divided into maximum blocksize blocks. After coding, the distortion between the reconstructed and original block is calculated. If that distortion fails to meet the predetermined threshold, the block being processed is subdivided into smaller blocks. The coding-testing procedure continues until the distortion is small enough or the smallest blocksize is reached.

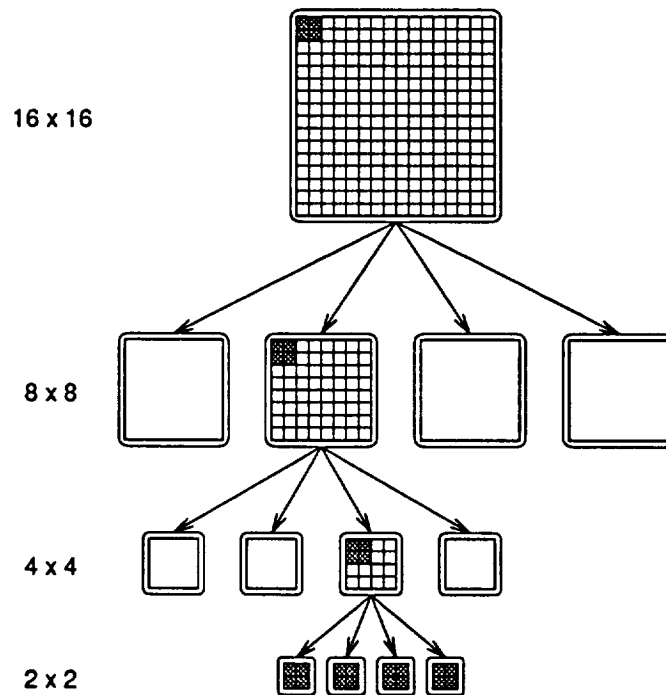
Mixture block coding with progressive transmission (MBCPT) is a coding scheme which combines MBC and progressive coding [6]. Progressive coding is an approach which uses successive approximations to converge to the target image, with the first approximation carrying “most” information and the following approximations enhancing it. In progressive coding, every pixel value, or the information contained in it, is possibly coded more than once and the total bit rate may increase due to different coding schemes

and quality desired.

With different stopping criteria, progressive coding is suitable for dynamic channel capacity allocation. If a predetermined distortion threshold is met, processing is stopped and no more refining action is needed. The threshold value can be adjusted according to the traffic condition in the channel. Successive approximations (or iterations) are sent through the channel in progressive coding and lead the receiver to the desired image. If these successive approximations are marked with decreasing priority, then a sudden decrease in channel capacity may only cause the received image to suffer from quality degradation rather than total loss of parts of the images. In Chapter 6, we will further address this issue in detail.

MBCPT is a multipass scheme in which each pass deals with different block sizes. The first pass codes the image with maximum block size and transmits it immediately. Only those blocks which fail to meet the distortion threshold go down to the second pass which processes the difference image block (coming from the original and coded image obtained in the first pass) with smaller blocks. The difference image coding process continues until the final pass which deals with the minimum size block. In our scheme, four passes (16x16, 8x8, 4x4, 2x2) are implemented. The quad tree structure, as shown in Figure 3.8, is adopted in our scheme. The 16x16 block is coded and the distortion of the block is calculated. If the distortion is greater than the predetermined threshold for 16x16 blocks, the block is divided into four 8x8 blocks for additional coding. This coding-checking procedure is continued until the only image blocks not meeting the threshold are those of size 2x2. After applying the discrete cosine transform, only four coefficients, including





**Figure 3.8** An example of quad-tree structure.

the DC and three lowest AC coefficients, are coded and others are set to zero. Because only partial blocks which fail to meet the distortion threshold need to be coded, side information is needed to instruct the receiver on how to reconstruct the image.

### 3.5.1 Simulation Results

Results obtained from the simulation show that substantial compression can be obtained while maintaining high image quality through the use of the MBCPT coding scheme. Table 3.6 shows the coding performance under different coding conditions. Figures 19-20 and 25-26 show the frame-by-frame performance for both sequences.  $T_1$  is the coding threshold for the decision mechanism in the quad tree. Maximum error in the block is

	Mean Rate (Mbits/sec)	STDR	PAR	Average PSNR	STDP	Subjective Test
Sequence 1	2.77	0.62	2.77	39.01	0.45	good with slightly blurry in background and neck areas
Sequence 2	2.63	0.64	3.01	38.85	0.47	slightly blocky in neck, but O.K.
Sequence 3	1.45	0.58	5.47	36.57	0.66	visible artifacts in neck, background regions
Sequence 4	3.68	0.95	2.14	36.37	0.80	extremely blocky, MC artifacts, unacceptable
Sequence 5	5.92	1.00	2.15	31.45	0.75	good
Sequence 6	4.10	0.94	3.10	28.81	0.95	slightly patchy, blocky but O.K.
Sequence 7	3.02	0.94	4.21	27.04	1.10	very patchy, visible artifacts, unacceptable
Sequence 8	8.21	1.66	1.55	31.43	0.76	slightly blurry, but O.K.

Sequence 1: "Susie", MC\_on, T1=10, T2=5

Sequence 2: "Susie", MC\_on, T1=10, T2=10

Sequence 3: "Susie", MC\_on, T1=15, T2=5

Sequence 4: "Susie", MC\_off, T1=10, T2=5\*

Sequence 5: "Football", MC\_on, T1=25, T2=25

Sequence 6: "Football", MC\_on, T1=35, T2=35

Sequence 7: "Football", MC\_on, T1=45, T2=45

Sequence 8: "Football", MC\_off, T1=25, T2=5\*

T1: Coding threshold for decision mechanism in quad tree (maximum error)

T2: Coding threshold after motion compensation (maximum error, \*: mean error)

**Table 3.6** Performance of coding rate and PSNR using MBCPT coding scheme.

Pass	Mean Rate	Standard Deviation	Maximum Rate	Minimum Rate
1 + overhead	253.48	51.06	530.93	178.28
2	288.32	40.47	665.85	203.12
3	932.62	129.81	1713.87	601.62
4	1300.16	492.52	4791.53	266.55
Total	2774.59	620.34	7702.19	1648.40

unit: Kbits/sec

**Table 3.7** Bit rate distribution among passes (*Susie*, MC\_on,  $T_1=10$ ,  $T_2=5$ ).

used as distortion measure. The same threshold has been used through out the coding process. However, different threshold combinations are possible to improve the overall quality/rate performance. For example, it is reasonable to set a large threshold for higher pass in the coding of full motion sequence since small details are not important in a fast-moving scene. On the other hand, a combination of large low pass and small high pass thresholds may be appropriate for smooth sequence like *Susie*.  $T_2$  is the coding threshold for 16x16 blocks after motion compensation. From Table 3.6, the average bit rate of Sequence 1 is 2.77 Mbits/s. The compression ratio is over 10 with a mean PSNR of 39.01 dB. This bit rate is higher than that of H.261 and ADTV schemes. However, as mentioned above, because the approach is based on a progressive coding scheme, this approach has some desirable properties for transmission over ATM networks. Figures 3.21, 3.27 and Table 3.7 show the frame-by-frame bit rate of four passes for Sequence 1. It is clear that the bit rate of the first three passes is relatively constant. However, the bit rate of pass 4 is bursty and highly uncorrelated. As pass 4 data is not essential to the reconstruction of the image, it can be transmitted with a lower priority for the total saving of bandwidth.

Figures 3.22 and 3.28 show the frame-by-frame PSNR of four passes for Sequence 1 and 5 respectively. Notice that the overall PSNR is quite constant which implies a substantial uniformity of quality. The small standard deviation of the PSNR agrees with subjective performance tests. Figures 3.23 and 3.29 shows the number of blocks for four different coding strategies as (w/ and w/o motion compensation, w/ and w/o coding). Different video contents and thresholds show different distribution. Figures 3.24 and 3.30 shows the percentage of coded blocks in four passes with chosen thresholds.

### 3.6 Some Notes

The overall quality performance depends heavily on the image activity. This fact is reflected by the lower PSNR of the full-motion *Football* sequence. A somewhat surprising phenomenon is noticed in the 40-70 frames of *Susie* sequence which presents more motion compared with the rest of the sequence. However, for a constant bit rate coding scheme, like H.261, this period has a higher PSNR value. On the other hand, for a variable bit rate coding scheme, like MBCPT, this period needs less coding resources to maintain a constant quality performance. This may be attributed to the function of motion compensation. From Figure 3.23, it is clear that more blocks in this period are motion compensatable. The motion compensation improves performance for about 2-3 dB on the global PSNR averages. All the coded sequences in this chapter are contained in an accompanying video tape for subjective comparison.

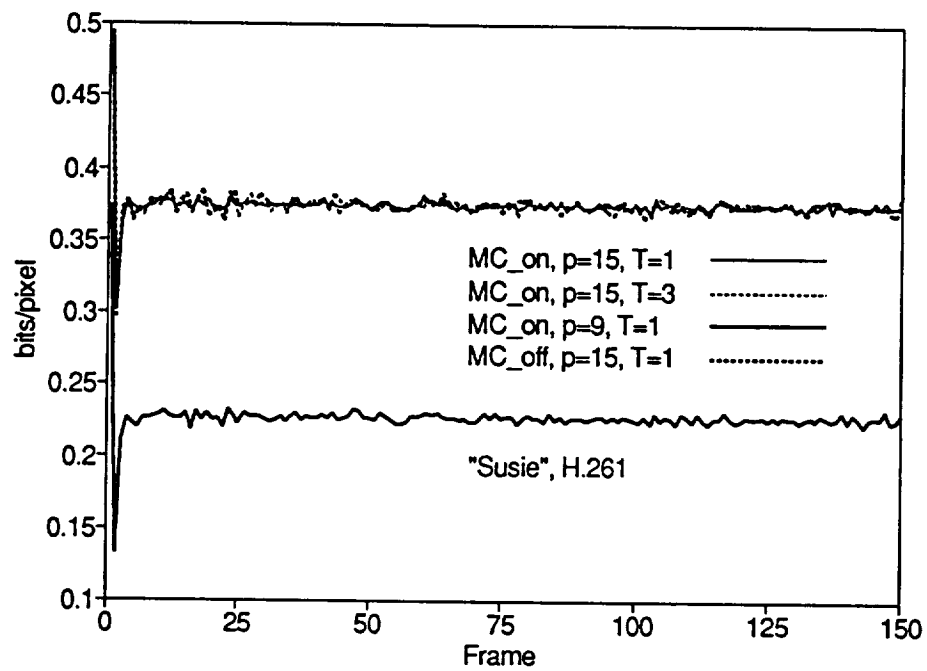


Figure 3.9 Coding rate of *Susie* sequence using H.261 algorithm.

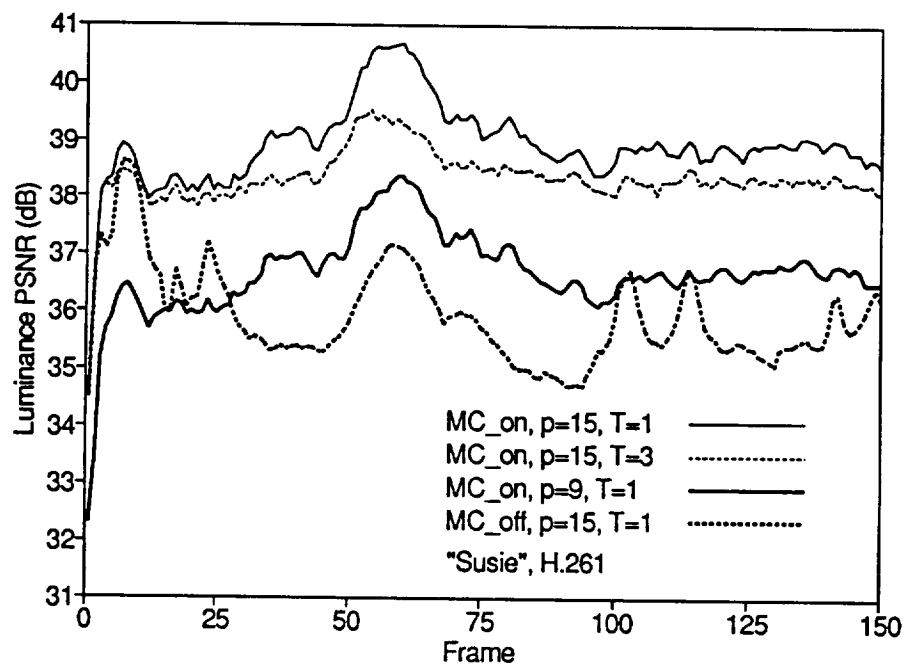


Figure 3.10 PSNR of *Susie* sequence using H.261 algorithm.

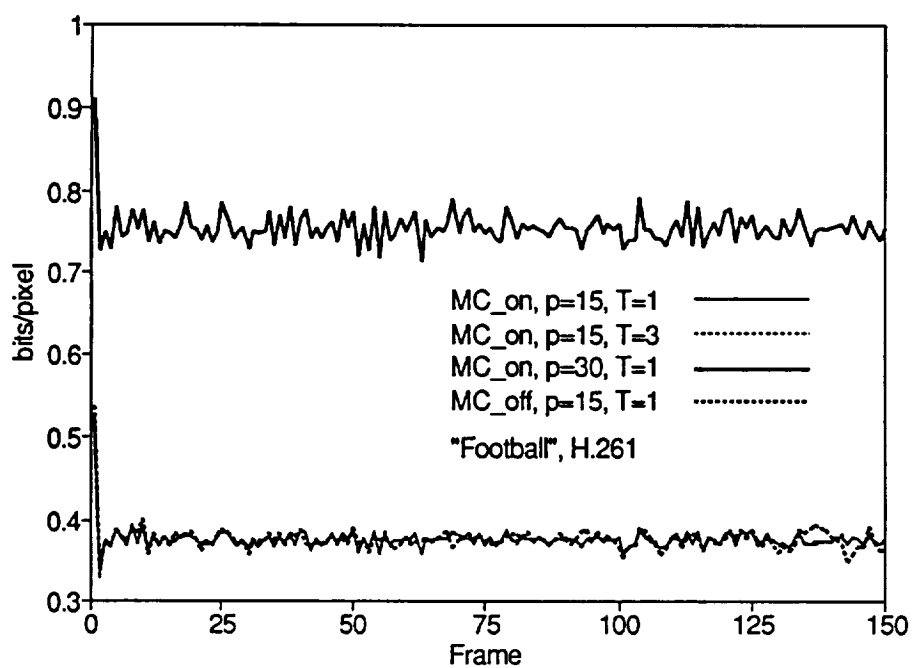


Figure 3.11 Coding rate of *Football* sequence using H.261 algorithm.

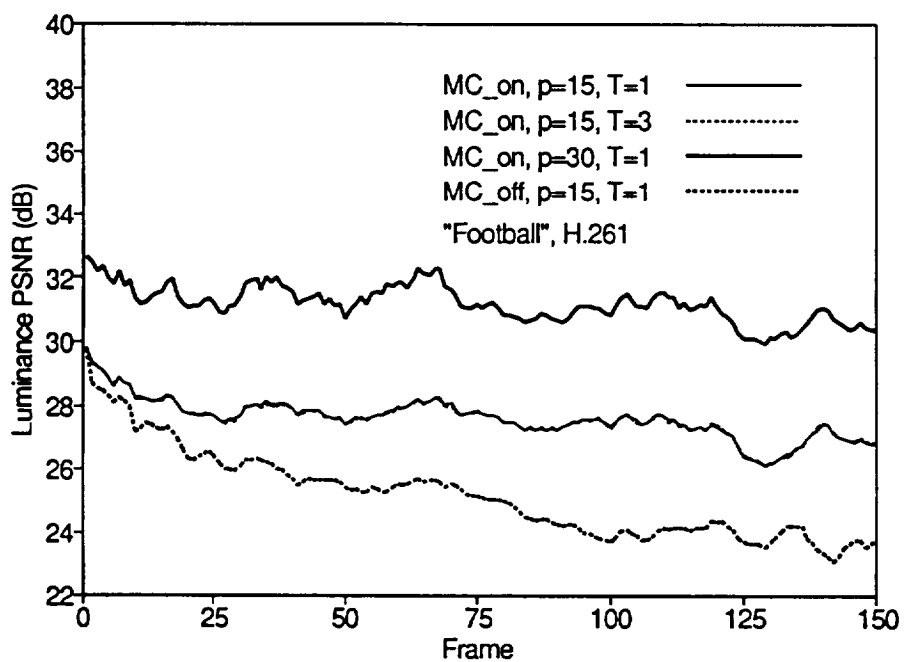


Figure 3.12 PSNR of *Football* sequence using H.261 algorithm.

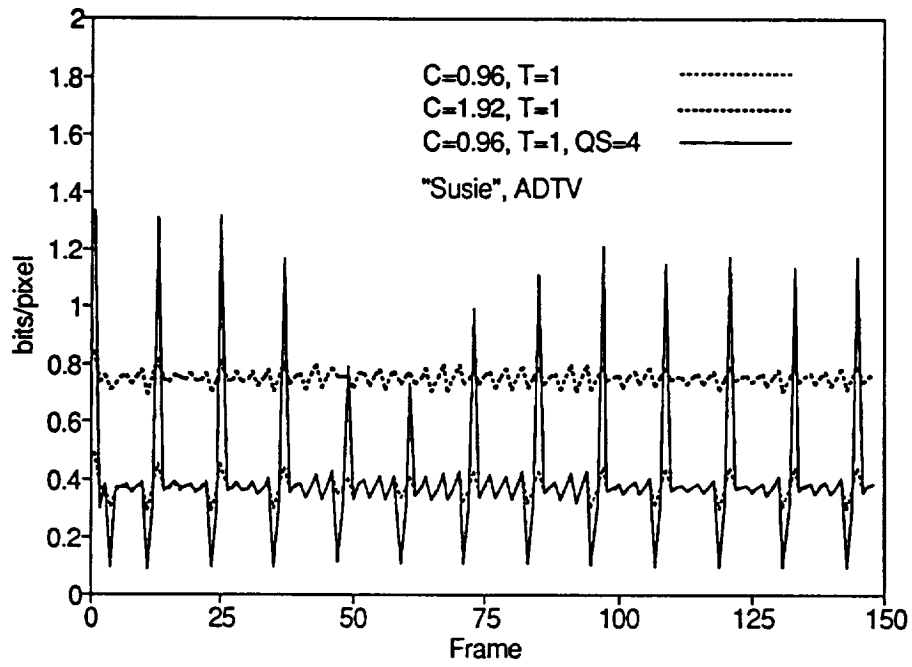


Figure 3.13 Coding rate of *Susie* sequence using ADTV algorithm.

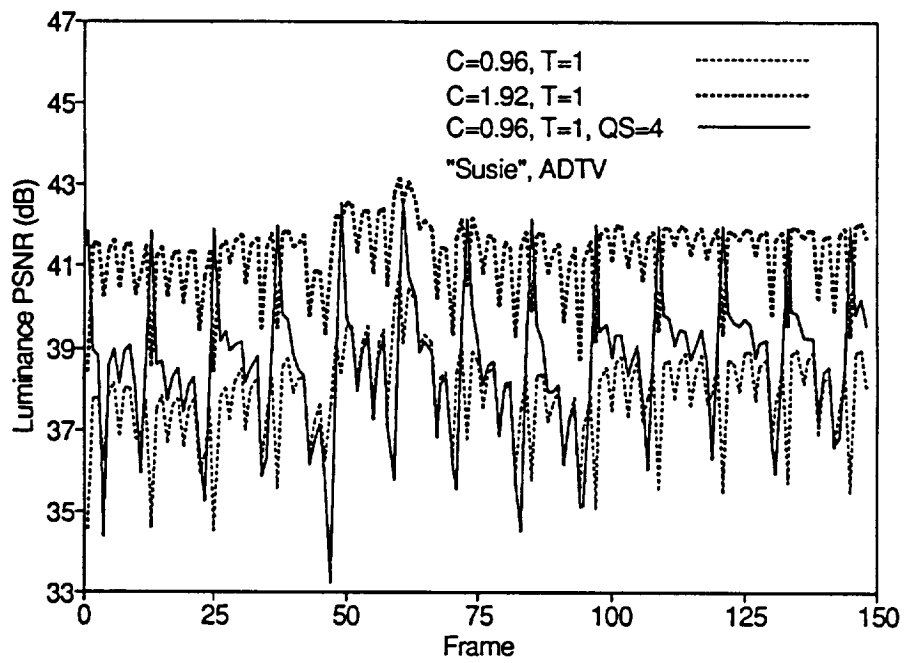


Figure 3.14 PSNR of *Susie* sequence using ADTV algorithm.

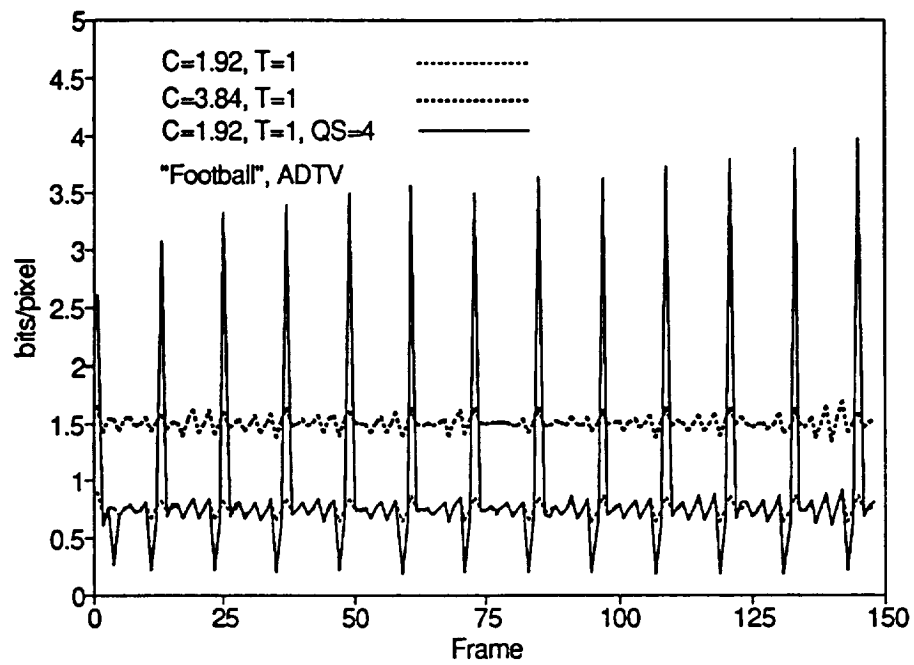


Figure 3.15 Coding rate of *Football* sequence using ADTV algorithm.

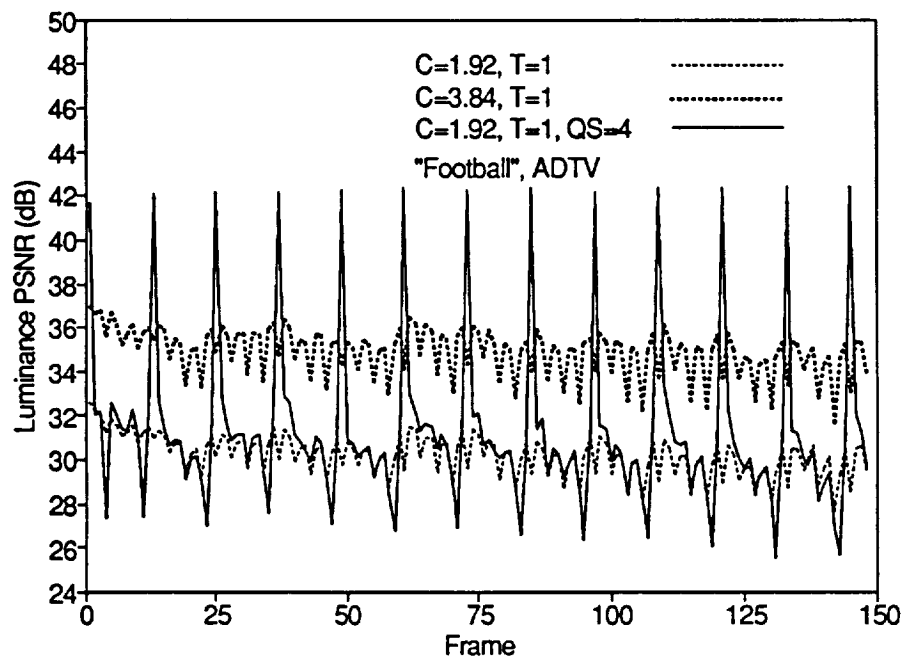


Figure 3.16 PSNR of *Football* sequence using ADTV algorithm.



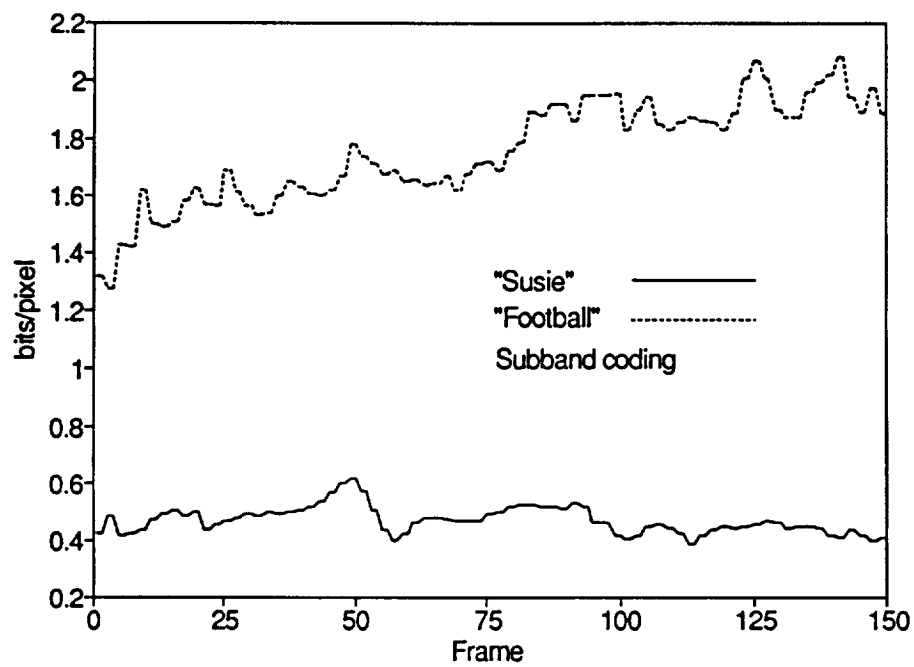


Figure 3.17 Coding rate using subband algorithm.

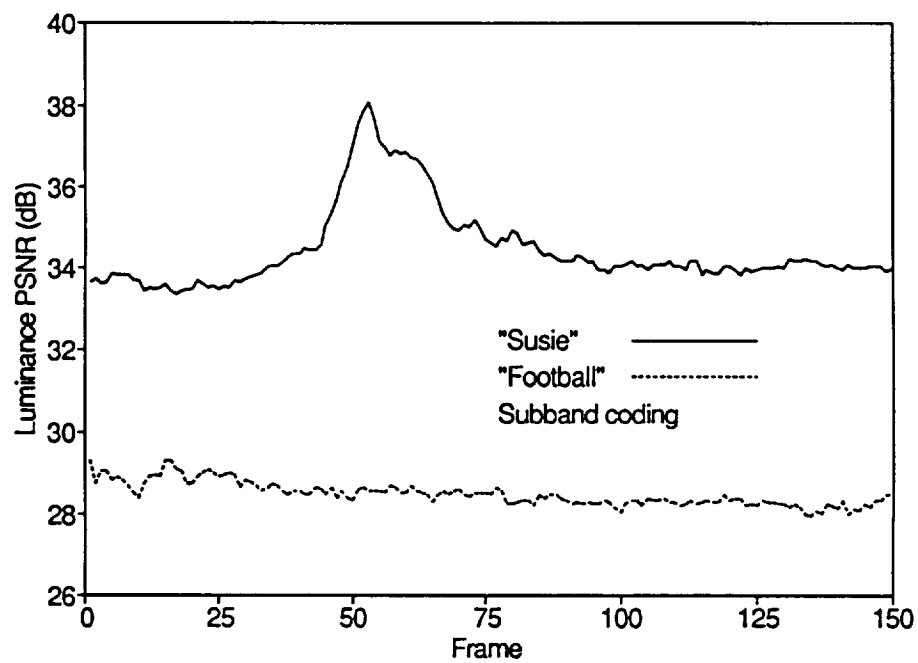


Figure 3.18 PSNR using subband algorithm.

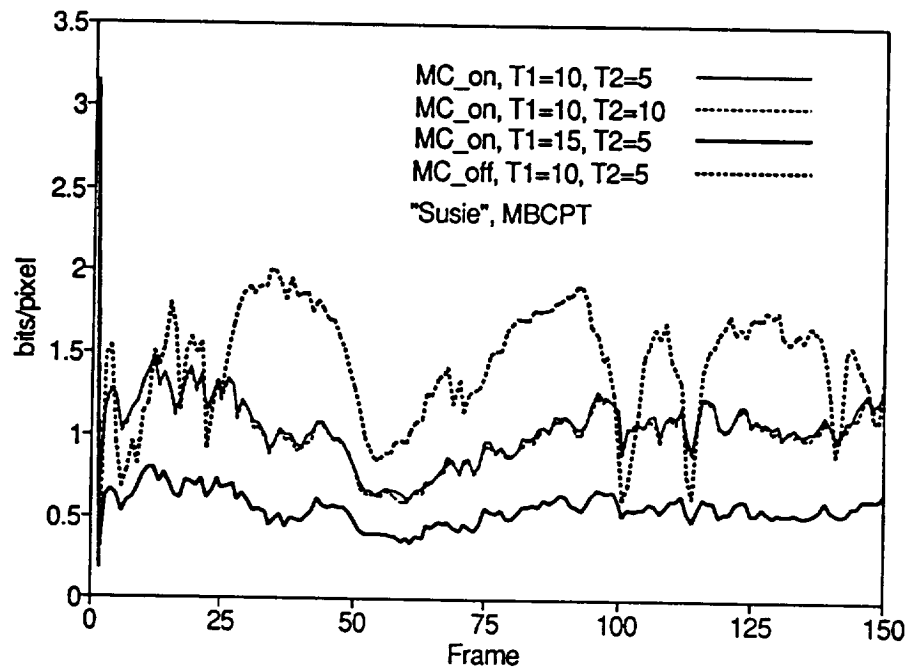


Figure 3.19 Coding rate of *Susie* sequence using MBCPT algorithm.

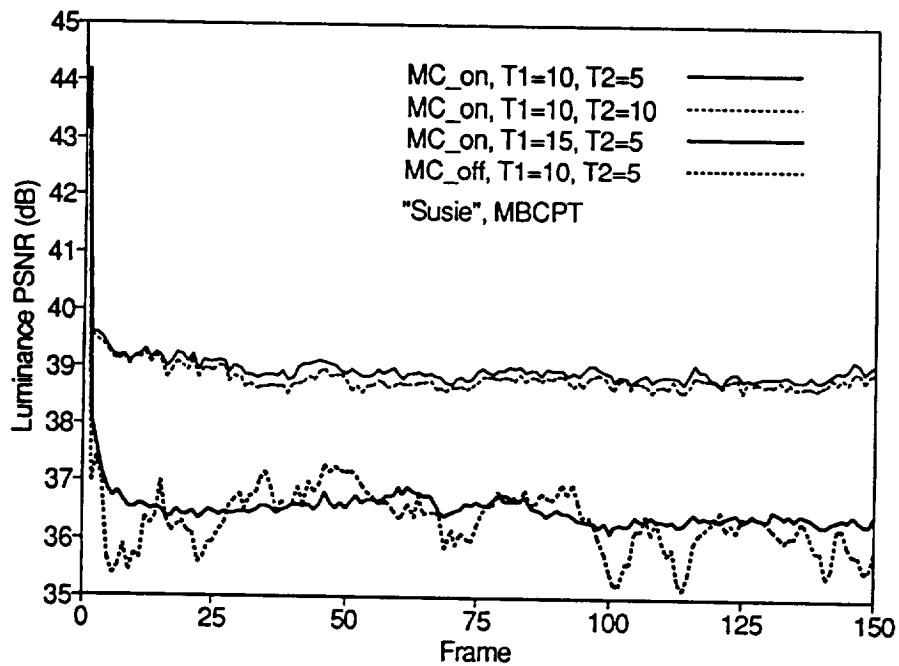


Figure 3.20 PSNR of *Susie* sequence using MBCPT algorithm.

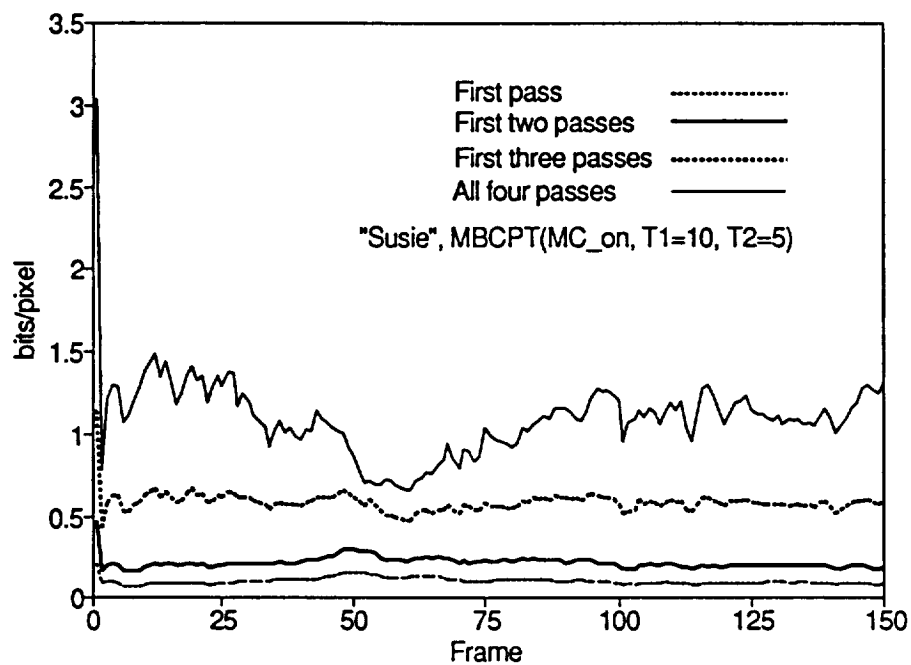


Figure 3.21 Coding rate distribution of four passes for *Susie* sequence.

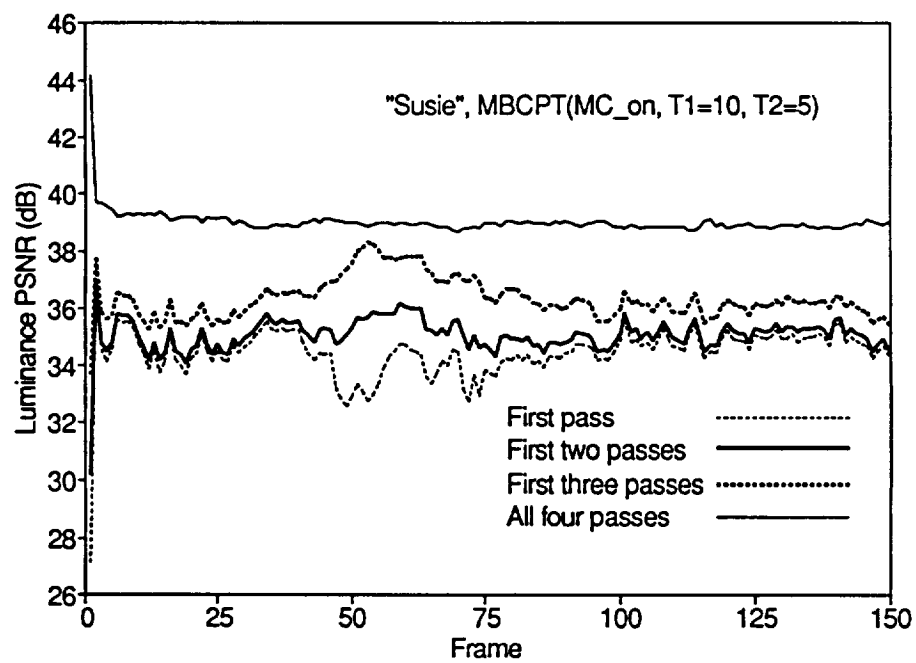


Figure 3.22 PSNR of four passes for *Susie* sequence.

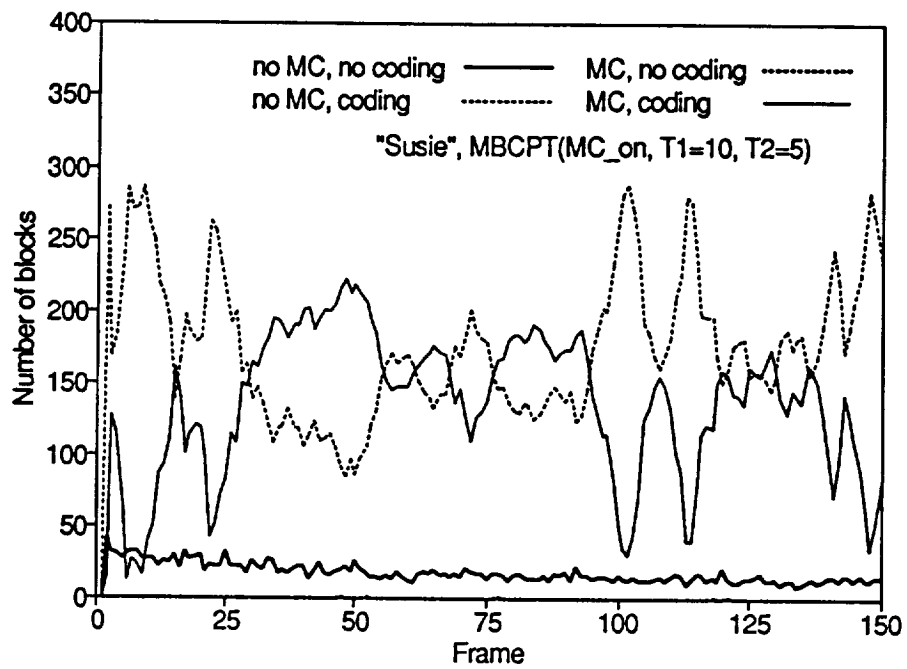


Figure 3.23 Number of blocks with different coding strategy (*Susie*).

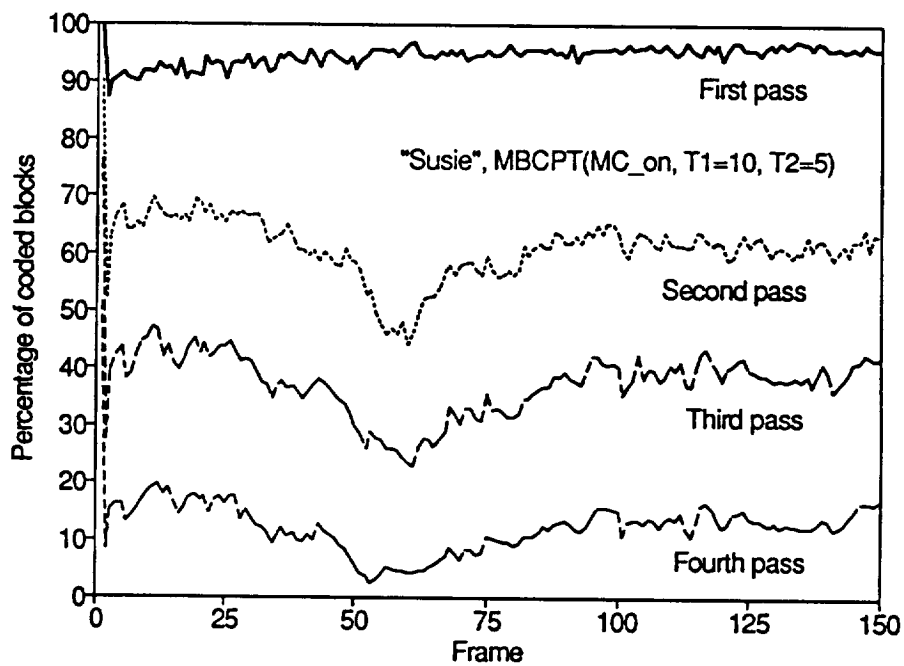


Figure 3.24 Percentage of coded blocks for four passes (*Susie*).

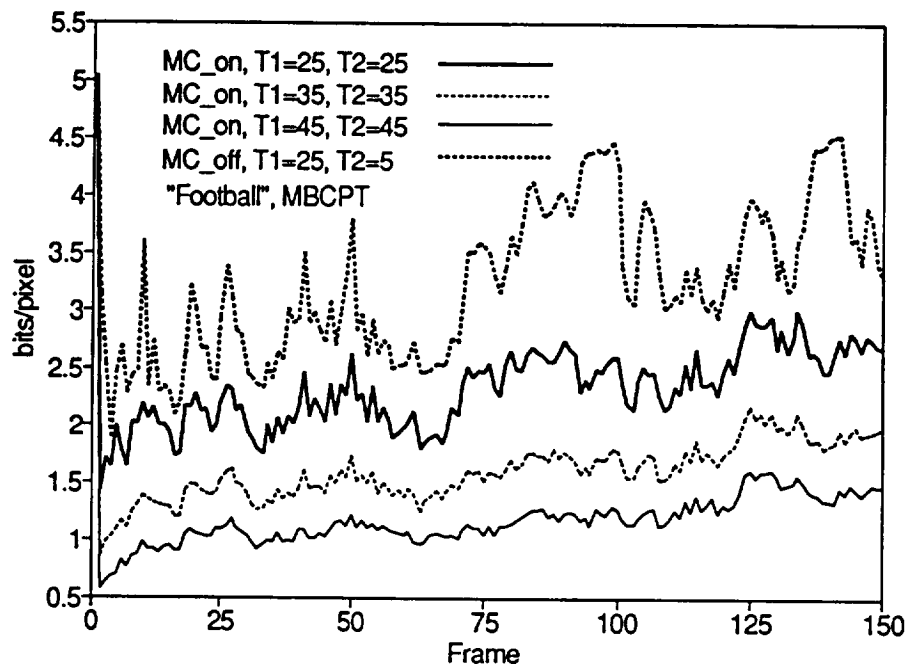


Figure 3.25 Coding rate of *Football* sequence using MBCPT algorithm.

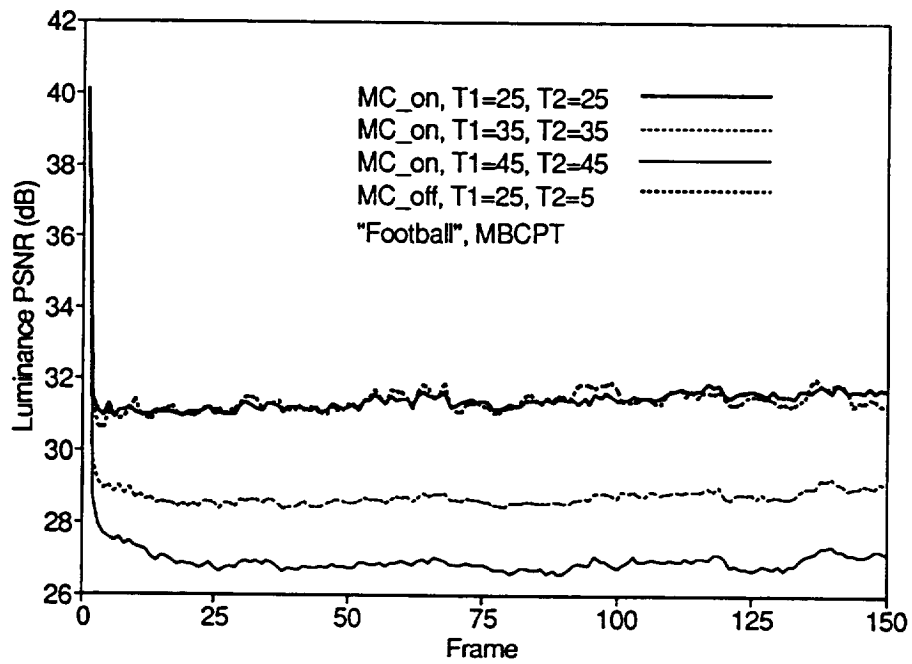
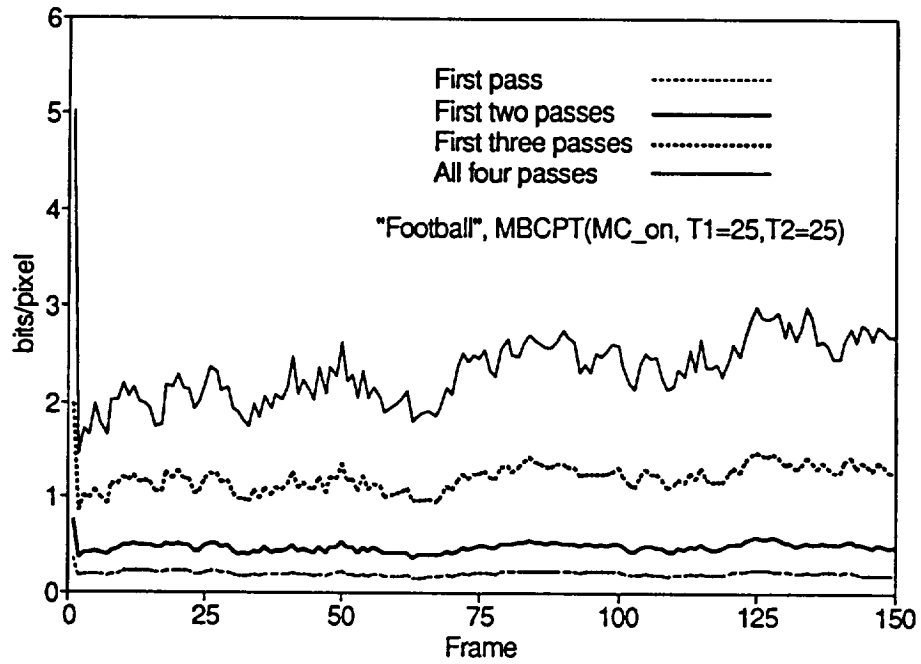
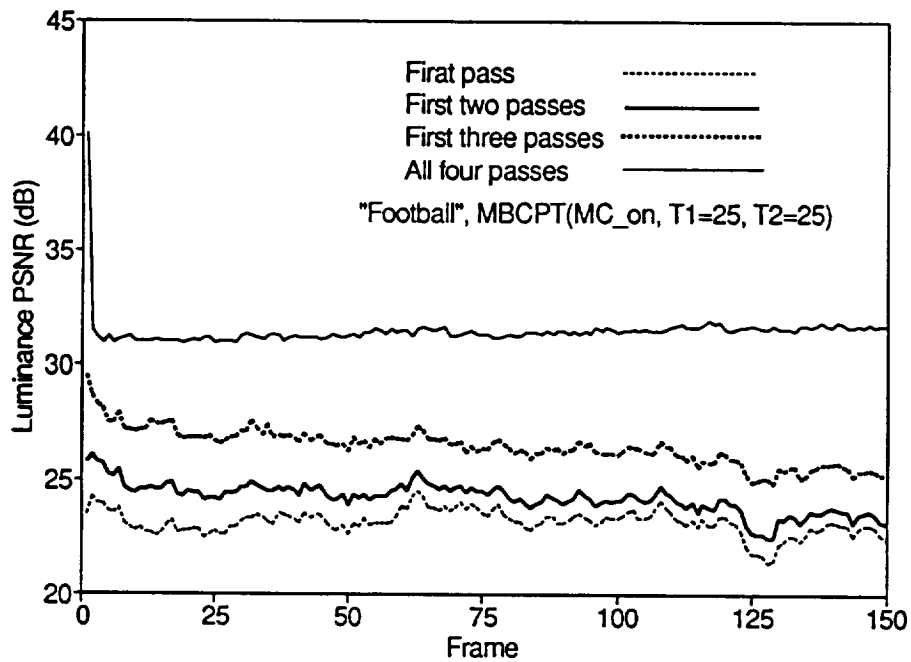


Figure 3.26 PSNR of *Football* sequence using MBCPT algorithm.



**Figure 3.27** Coding rate distribution of four passes for *Football* sequence.



**Figure 3.28** PSNR of four passes for *Football* sequence.

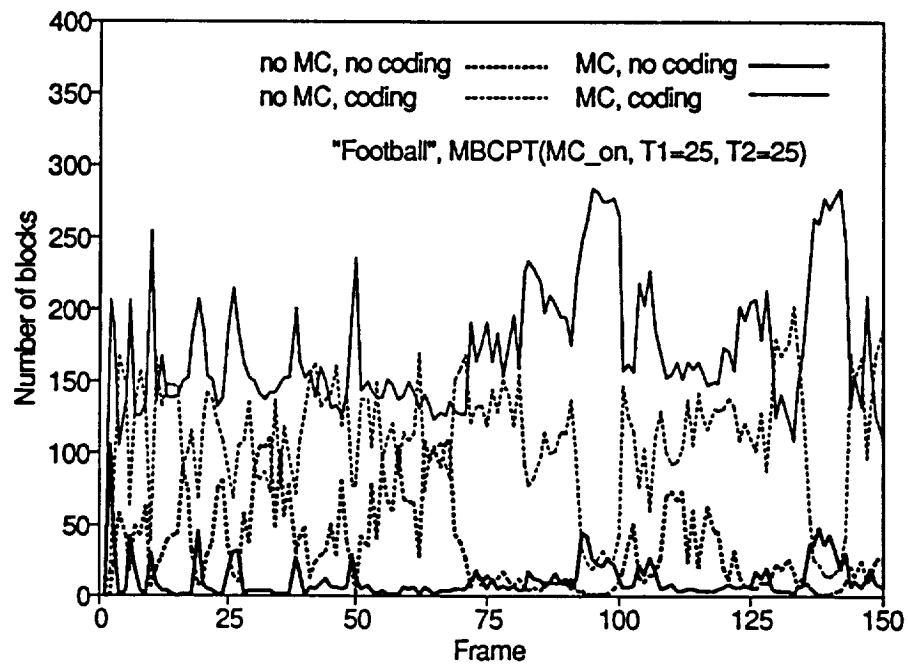


Figure 3.29 Number of blocks with different coding strategy (*Football*).

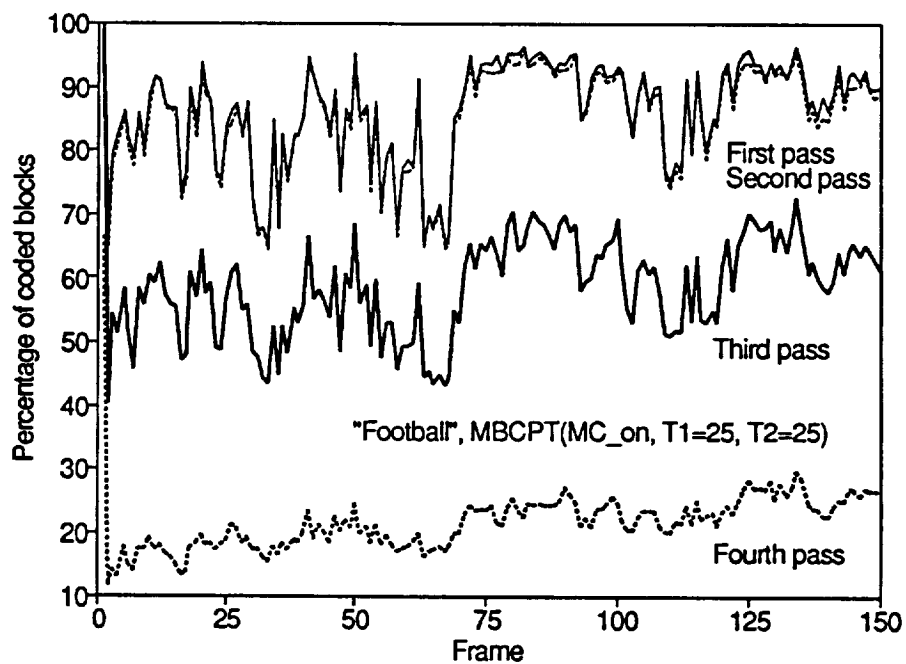


Figure 3.30 Percentage of coded blocks for four passes (*Football*).

62

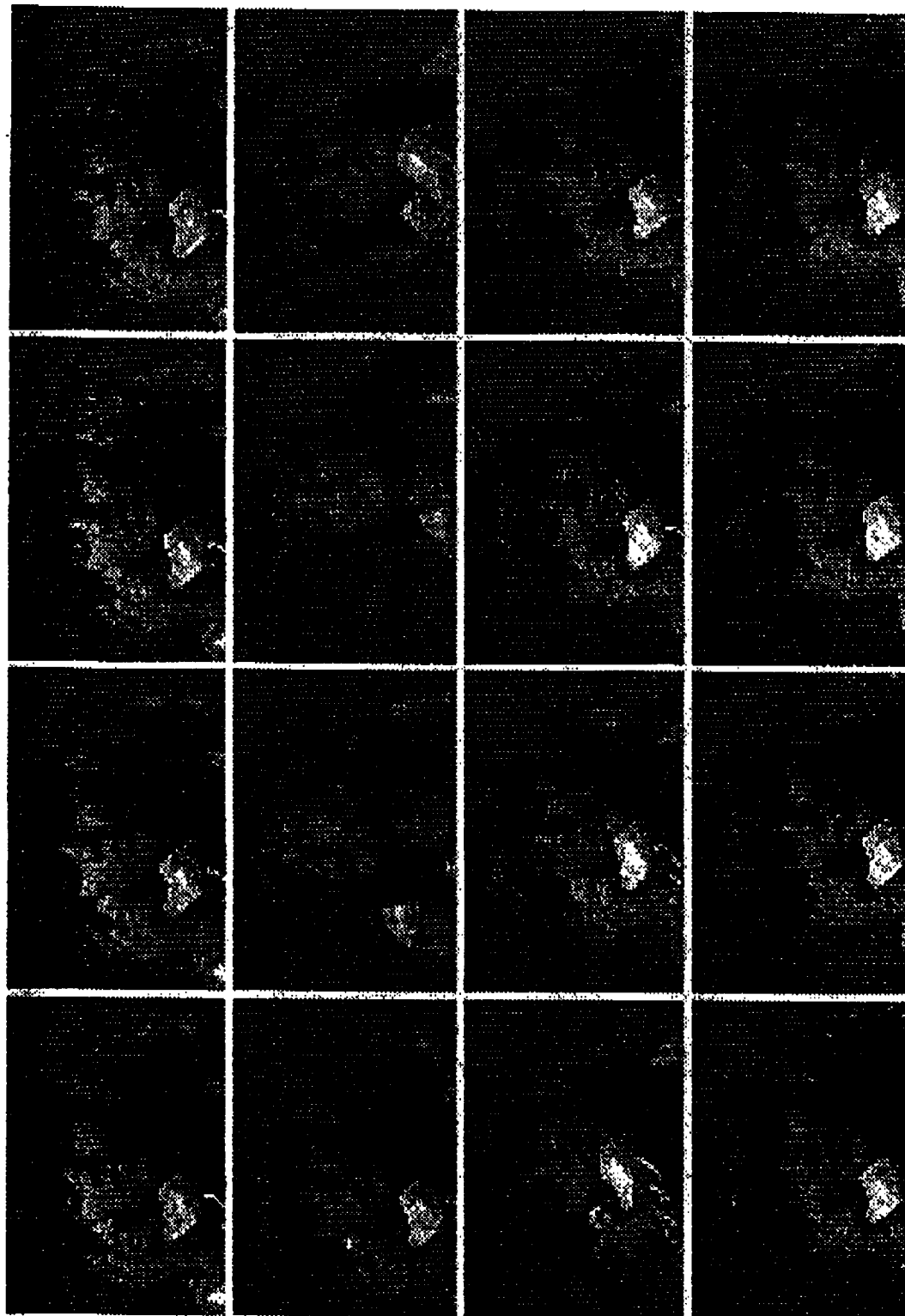


Figure 3.31 Susie sequence (original, every tenth frame, left to right, top to bottom).



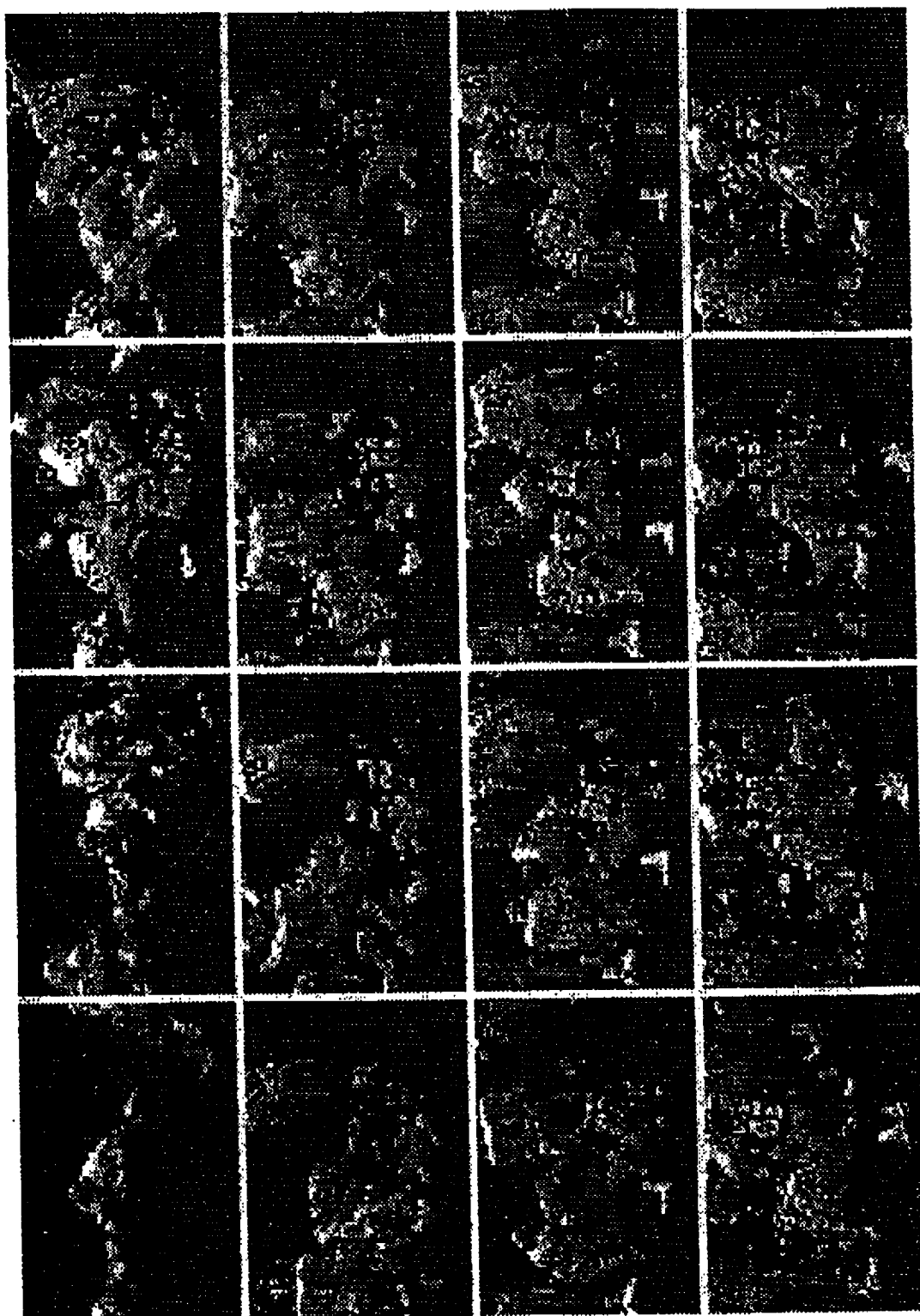


Figure 3.32 Football sequence (original, every tenth frame, left to right, top to bottom).

## Chapter 4

# Video Source Modelling

In the previous chapters we have introduced some of the basic definitions and concepts in B-ISDN, and the video coding algorithms that we will be using in this work. Starting from this chapter we begin to focus more on the interface between the network and the video coder. In this chapter we will develop different models for the video source. While modelling of the video source is not a new concept, models currently available in the literature basically deal with one type of video source. In this chapter, we test the validity of models with two extreme sequences which video sources may present. Besides, we do not only test the statistical fitness of proposed models, we also examine their queueing performance in the network considering the effects of packetization. The reason modelling is important to our work is because performance simulation is very critical when designing a coding scheme which will hopefully best fit into the future ATM environment. Efficient and accurate simulation depends on accurate modelling. Unfortunately, modelling of a video source is more difficult when compared to a speech source since video is a highly complex source. When developing a model, one has to

decide on the degree of complexity and sophistication required of the model. Too much attention to detail can lead to models that are mathematically or computationally intractable. Our objective in modelling the video source is relatively modest. We would like to match the second order statistics of the data. This model can then possibly be used for prediction of rate fluctuations as well as in simulation.

Various video source models have been proposed. For simulation purposes, the source is modelled with a first order AR process by Nomura *et al.* [23] and Maglaris *et al.* [24]. A Markov chain model was proposed by Heyman *et al.* [22]. For queueing analysis, Maglaris *et al.* [24] and Sen *et al.* [25] model the source as a birth-death Markov process. A more elaborate model which combines the first-order AR process and a three state Markov chain was presented by Ramamurthy *et al.* [30]. Melamed *et al.* [33] model a VBR video source by exploiting bit-rate histograms and autocorrelation functions. In the following sections, we develop some models to simulate the video coding rates which were obtained in the last chapter. We will also validate these models by performing the goodness-of-fit tests.

## 4.1 Video Source Sequences

As mentioned above, different types of video sources generate sequences with different statistical characteristics, and different bit rates. In this chapter, two sequences with very different characteristics, namely *homogeneous* and *scene-cut* sequences, will be used to explore the capability of models. In order to increase the validity of sample data, the homogeneous sequence (600 frames) consists of four MBCPT coded *Susie* sequences with

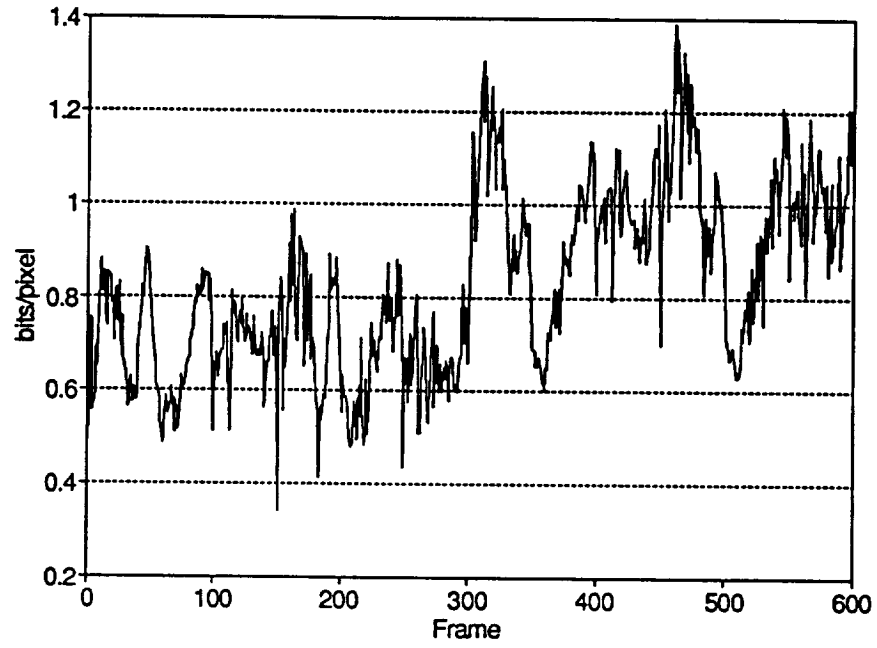


Figure 4.1 Coding bit-rate of Sequence 1.

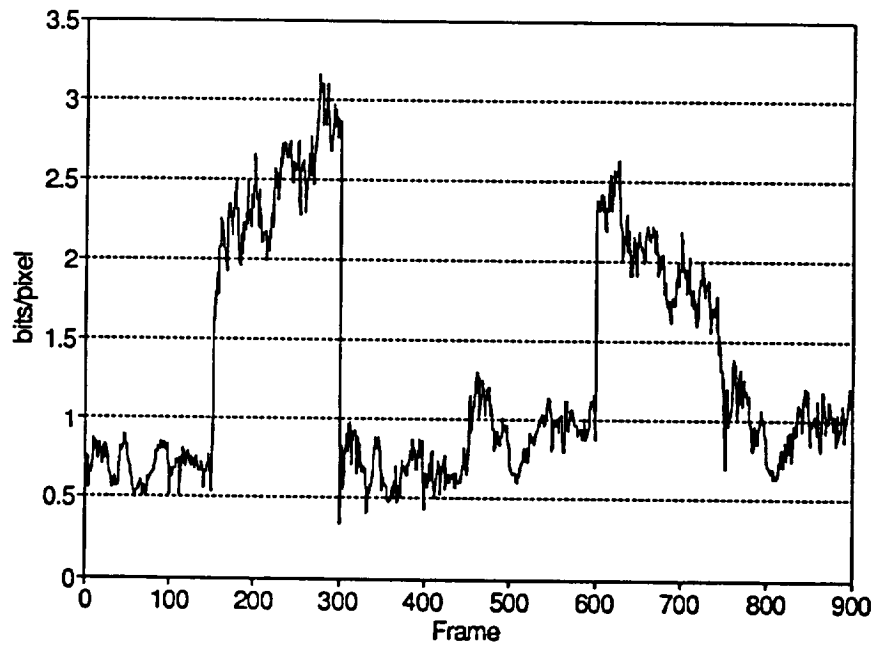


Figure 4.2 Coding bit-rate of Sequence 2.

various thresholds. MBCPT coded data is chosen because it is more bursty. The *Susie* sequence, as shown in Figure 3.31, shows a women talking on the phone. Therefore, the homogeneous sequence can represent videophone/videoconference type video without abrupt motion, hence the name homogeneous. For the scene-cut sequence, four *Susie* and two *Football* sequences are mixed together to represent broadcast type video. The *Football* sequence, as shown in Figure 3.32, presents full-motion football action which is sharply contrary to the *Susie* sequence. This scene-cut sequence features four scene cuts which occur at frame 150, 300, 600 and 750, as shown in Figure 4.2. Figure 4.1 shows the bit rate (averaged for each frame) of the homogeneous sequence for all 600 frames (20 seconds). The average value  $\mu$  over all 600 frames and the standard deviation  $\sigma$  were found to be  $\mu = 0.82$  bits/pixel and  $\sigma = 0.1958$  bits/pixel. The maximum value of the bit-rate is 1.39 bits/pixel and the minimum value of the bit-rate is 0.33 bits/pixel. Figure 4.2 shows the bit rate for the scene-cut sequence. The mean value  $\mu$  and the standard deviation  $\sigma$  are 1.29 and 0.7081 bits/pixel respectively. The maximum value of the bit-rate is 3.15 bits/pixel and the minimum value of the bit-rate is 0.33 bits/pixel. For simplicity, these two sequences are referred to as Sequences 1 and 2 in the rest of this chapter.

## 4.2 Models for Homogeneous Sequences

We first develop models for Sequence 1. From Figure 4.1, we can see that there are not too many discontinuities or very steep changes in the bit rate. Given the homogeneous nature of the sequence we looked at the following modelling approaches

- Continuous state autoregressive Markov model
- Discrete time Markov chain model
- Discrete state, continuous time birth-death Markov model

#### 4.2.1 Continuous State Autoregressive Markov Model

An autoregressive process of order  $p$  (or  $AR(p)$ ) can be expressed as

$$X_n = \sum_{i=1}^p a_i X_{n-i} + Z_n \quad (4.1)$$

where  $Z_n$  is a sequence of white noise. Our aim is to find the coefficients vector  $\mathbf{A} = (a_1, \dots, a_p)$  and the white noise variance  $\sigma^2$  based on the bit-rate  $\lambda_1, \dots, \lambda_n$ . For a stationary AR process, the coefficients of the AR model are related to the autocorrelation coefficients by the *Yule-Walker* equations [28]. PEST is a software package for the estimation and analysis of time series [28]. In PEST, the *Yule-Walker* estimator is used to obtain the preliminary estimation for an AR model. The preliminary estimated model is then optimized using a *maximum likelihood* estimator. Using the PEST, we first obtained AR(1) as follows

$$X_n = 0.829 + 0.907X_{n-1} + Z_n, \quad Z_n \sim WN(0, 0.007042) \quad (4.2)$$

When an AR model is fitted to a given series, an essential part of the procedure is to examine the residuals, which should, if the model is satisfactory, have the appearance of white noise. If the autocorrelations and partial autocorrelations of the residuals suggest that they come from some other identifiable process, then some other models are recommended. To test for independence in the residuals of this AR(1) model, several

randomness tests in PEST were applied. These includes the *Portmanteau* test, the *turning-Point* test, the *difference-Sign* test and the *rank* test. Test results show that the residuals from this AR(1) model are not white. We therefore increase the order of AR model. Figure 4.3 shows that the autocorrelation function of AR(4) model has a better match for the empirical autocorrelation. The Yule-Walker and maximum-likelihood estimator provide the following AR(4) model

$$X_n = 0.829 + 0.770X_{n-1} - 0.021X_{n-2} + 0.093X_{n-3} + 0.096X_{n-4} + Z_n, \quad Z_n \sim WN(0, 0.006698) \quad (4.3)$$

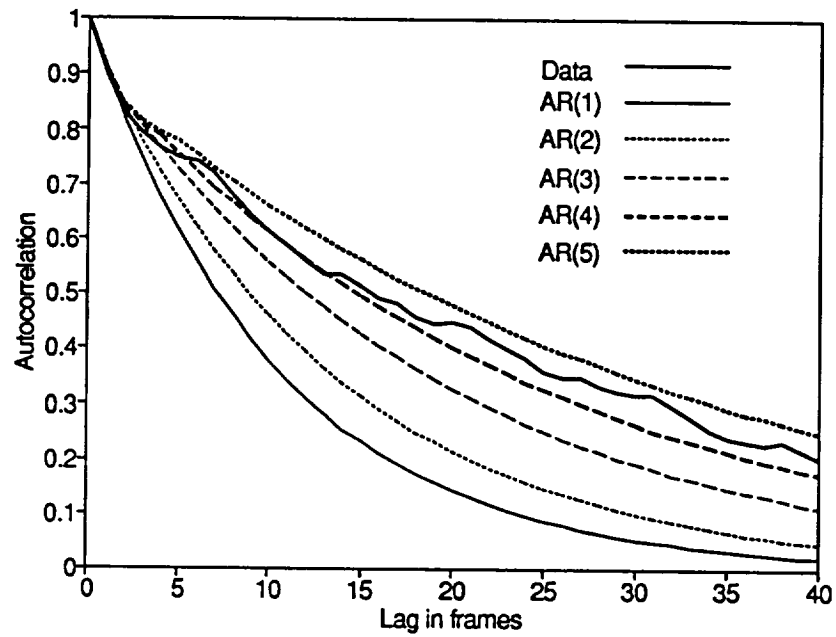
We again tested the residuals using the four tests listed above. Based on the satisfactions of above tests, we concluded that AR(4) was a suitable model for Sequence 1.

#### 4.2.2 Discrete Time Markov Chain Model

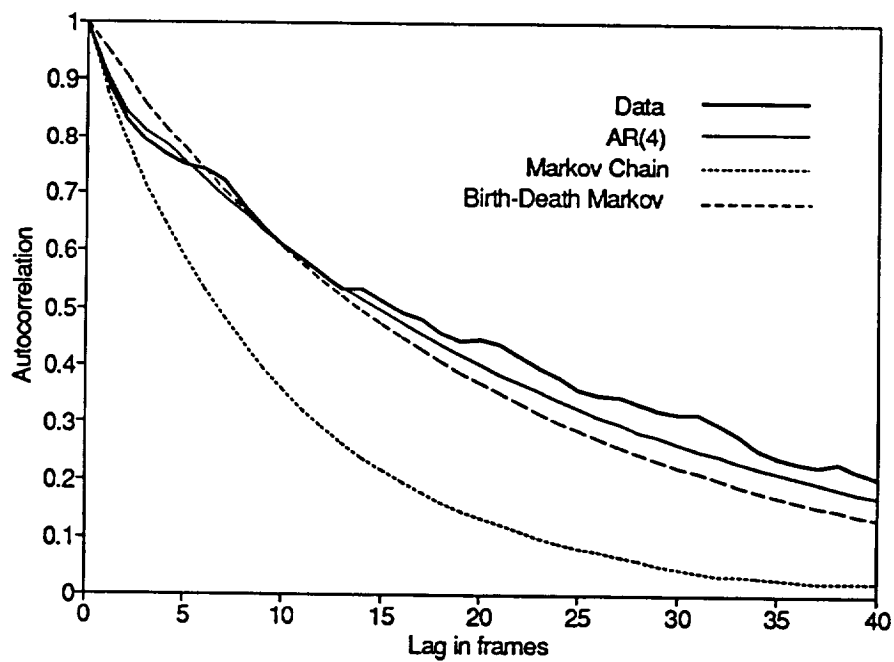
A Markov process with a discrete state space is referred to a Markov chain. To set up the model, the bit-rate  $\lambda_n$  was quantized into discrete levels using a uniform quantizer with stepsize 0.05 bits/pixel. Each quantization level is a state of the Markov chain model. In the simulation, the number of states is chosen as 22 in order to cover the range of bit-rate values. With Sequence 1, one-step transition probability  $r_{ij}$  is estimated in the usual way:

$$r_{ij} = \frac{\text{number of transitions from } i \text{ to } j}{\text{number of transitions out of } i} \quad (4.4)$$

when the denominator is greater than zero. When the denominator is zero,  $r_{ii}$  is set to be 1. Since there is no transition out of state  $i$ , this assignment will not affect the stationary distribution. The autocorrelation function of the Markov chain model, as shown in Figure



**Figure 4.3** Autocorrelation functions of Sequence 1 and AR(1)-AR(5).



**Figure 4.4** Autocorrelation functions of several models (Sequence 1).



4.4, does not match that of Sequence 1 very well. Accuracy can be improved by refining the quantization stepsize.

#### 4.2.3 Discrete State, Continuous Time Birth-Death Markov Model

In this section, the possibility of modelling “smooth” sequences with a birth-death Markov model, which has been proposed in [24], is investigated using Sequence 1. Unlike the Markov chain model, the birth-death Markov model allows only transitions between neighboring states which reflects the fact that there won’t be any sudden changes in the process. Based upon observations, it is assumed that there exists a tendency of the bit-rate toward higher levels to decrease at high levels, and inversely, the tendency of the bit-rate toward lower levels to increase at high levels. This is a reasonable assumption and will result in a bell shaped stationary distribution of the state. In the model, the bit rate is quantized with uniform quantization step  $A$  bits/pixel, and  $M + 1$  possible levels,  $(0, A, \dots, MA)$ . The transition rates  $r_{i,j}$  from state  $iA$  to  $jA$  are given by

$$\begin{aligned} r_{i,i+1} &= (M-i)\alpha & i < M \\ r_{i,i-1} &= i\beta & i > 0 \\ r_{i,i} &= 0 \\ r_{i,j} &= 0 & |i-j| > 1 \end{aligned} \tag{4.5}$$

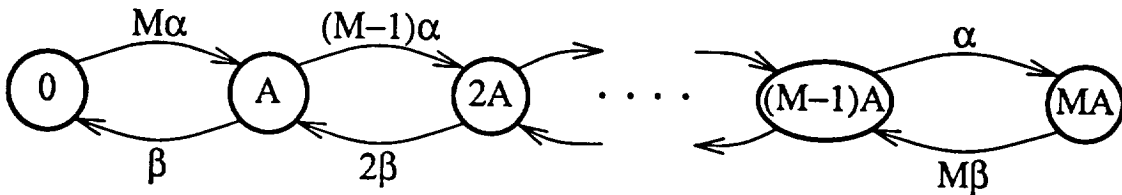
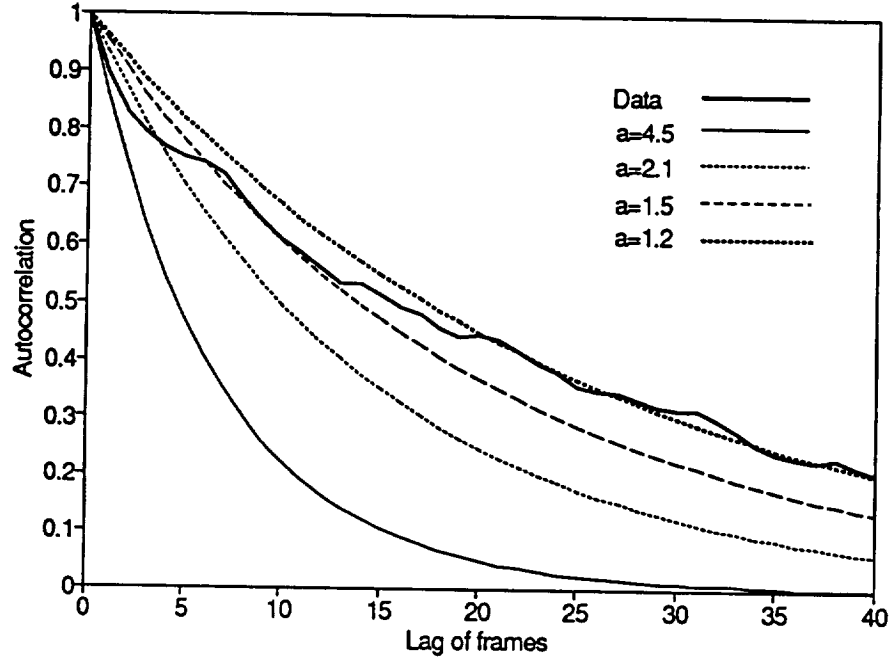


Figure 4.5 State-transition-rate diagram for birth-death Markov model.



**Figure 4.6** Exponential fits ( $e^{-a\tau}$ ,  $\tau = \text{lag}/30$ ) of autocorrelation function (Sequence 1).

Figure 4.5 shows such a birth-death process. It is easy to show [29] that  $\lambda_n$  at steady state has a binomial distribution with mean  $E(\lambda)$ , variance  $C(0)$  and exponential autocovariance  $C(\tau)$

$$\begin{aligned}
 P\{\lambda_n = kA\} &= \binom{M}{k} p^k (1-p)^{M-k} \\
 p &= \frac{\alpha}{\alpha + \beta} \\
 E(\lambda) &= MAp \\
 C(0) &= MA^2 p(1-p) \\
 C(\tau) &= C(0)e^{-(\alpha + \beta)\tau}
 \end{aligned} \tag{4.6}$$

The parameters of this model  $A$ ,  $\alpha$ ,  $\beta$  can be estimated by matching the above equations with the measured values and take  $M$  as a parameter. Note that the autocorrelation function is fitted by an exponential equation of the form  $C(\tau)/C(0) = e^{-a\tau}$  by sampling at

30 frames/s, where  $\tau$  is an unit in second and  $a$  is the best matching coefficient. From Figure 4.6, we take  $a = 0.05/(1/30) = 1.5$ . In this binomial model, the rate  $\lambda_n$  can be thought of as the aggregate rate from  $M$  independent minisources, each alternating between transmitting 0 (the OFF state) and  $A$  bits/pixel (the ON state) as shown in Figure 4.7.

We assume a continuous state queue which is fed by  $M$  independent minisources, each sending  $\lambda$  units of flow per time unit when it is on. The queue empties with fixed rate  $\mu$  units of flow per time unit, also shown in Figure 4.7. A two dimensional state  $\{q(t), k(t)\}$  is built to completely describe the queueing system. The first component represents the length of queue at time  $t$ , while the second component shows how many minisources are on at that instant. Figure 4.8 shows the state-transition-rate diagram and Figure 4.9 shows the instantaneous transition rate matrix  $Q$ , for  $M$  equals 3.  $Q$  can be represented in another form with defined  $A, B, C$ . We define  $\pi = [\pi_0, \pi_1, \pi_2, \dots]$  as the limiting state probability where  $\pi_i = [\pi_{i0}, \pi_{i1}, \dots, \pi_{iM}]$ ,  $\pi_{ij}$  is the limiting probability of the state with queue length  $i$  and there are  $j$  minisources on. For an ergodic Markov chain, we can express the equilibrium solution, through the *Chapman-Kolmogorov* equation, in matrix form as

$$\pi Q = 0 \quad (4.7)$$

This last equation coupled with the probability conservation relation, namely

$$\sum_i \pi_i = 1 \quad (4.8)$$

uniquely gives us the limiting state probabilities. With  $Q$  expressed as in Figure 4.9, we

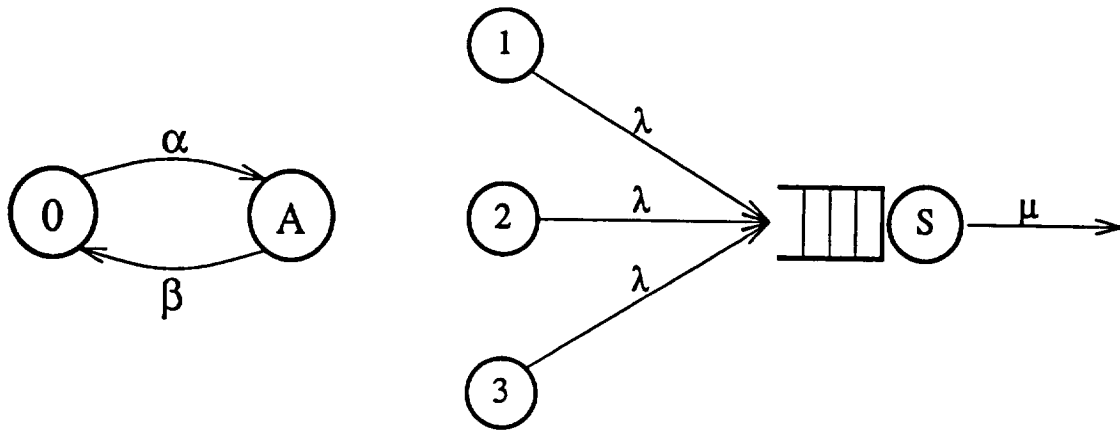


Figure 4.7 Queueing system for minisource model ( $M = 3$ ).

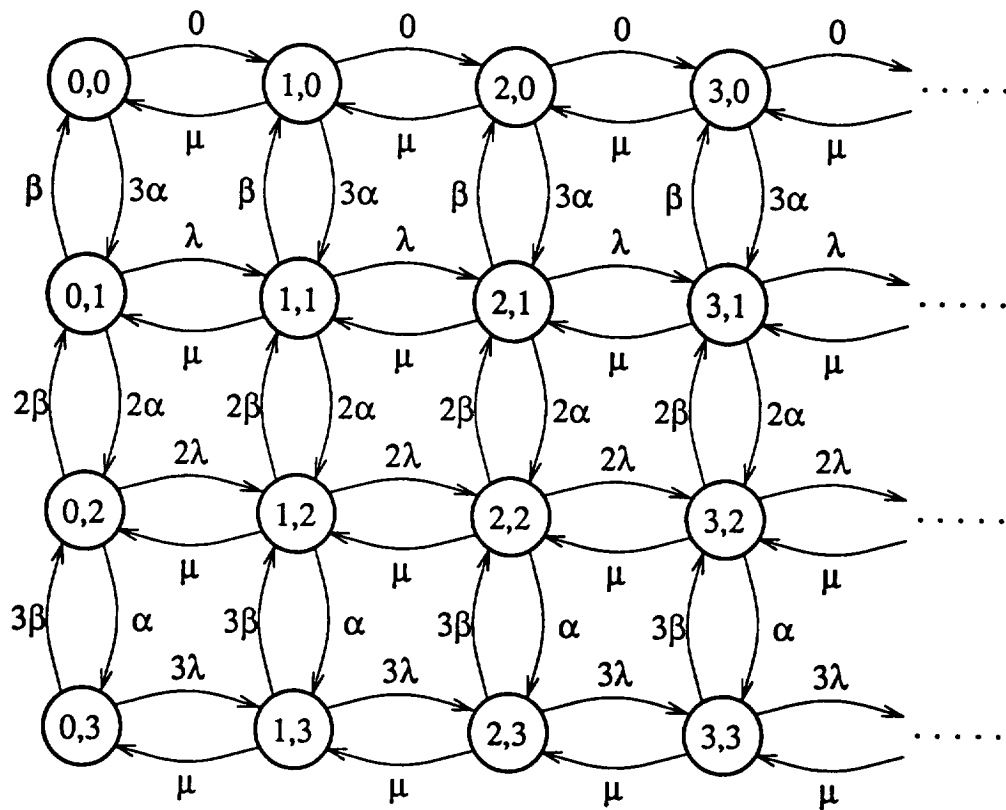


Figure 4.8 State-transition-rate diagram for minisource ( $M = 3$ ).

	00	01	02	03	10	11	12	13	20	21	22	23
00	$-3\alpha$	$3\alpha$			0							
01	$\beta - (\lambda + 2\alpha + \beta) 2\alpha$					$\lambda$						
02	$2\beta - (2\lambda + \alpha + 2\beta) \alpha$						$2\lambda$					
03	$3\beta - (3\lambda + 3\beta)$							$3\lambda$				
10	$\mu$				$-(3\alpha + \mu)$	$3\alpha$			0			
11		$\mu$			$\beta - (2\alpha + \beta + \lambda + \mu) 2\alpha$					$\lambda$		
12			$\mu$		$2\beta - (\alpha + 2\beta + 2\lambda + \mu) \alpha$						$2\lambda$	
13				$\mu$	$3\beta - (3\beta + 3\lambda + \mu)$							$3\lambda$
		$\vdots$				$\vdots$				$\vdots$		

$$A = \begin{bmatrix} -3\alpha & 3\alpha \\ \beta - (\lambda + 2\alpha + \beta) 2\alpha \\ 2\beta - (2\lambda + \alpha + 2\beta) \alpha \\ 3\beta - (3\lambda + 3\beta) \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \lambda \\ 2\lambda \\ 3\lambda \end{bmatrix}$$

$$C = \begin{bmatrix} \mu \\ \mu \\ \mu \\ \mu \end{bmatrix}$$

$$Q = \begin{bmatrix} A & B & 0 & \dots \\ C & A-C & B & \dots \\ 0 & C & A-C & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Figure 4.9 Instantaneous transition rate matrix of minisource model.

have

$$\begin{aligned} \pi_0 A + \pi_1 C &= 0 & \text{for } i = 0 \\ \pi_{i-1} B + \pi_i (A - C) + \pi_{i+1} C &= 0 & \text{for } i \geq 1 \end{aligned} \quad (4.9)$$

Many methods are available for solving this set of equations, for example the method of z-transforms. Here, we just use a simple computational approach. For a stationary process,

we assume

$$\pi_{i+1} = \pi_i R \quad \text{for } i \geq 1 \quad (4.10)$$

Substitute Eq. (4.10) into the second equation of Eq. (4.9), we obtain

$$B + R(A - C) + R^2 C = 0 \quad (4.11)$$

Rearranging the last equation, we have

$$R = -(R^2 C + B)(A - C)^{-1} \quad (4.12)$$

We can solve  $R$  simply by using computer iteration with initialization of  $R$  to be the null matrix. Applying Eq. (4.8) and the first equation of Eq. (4.9) we obtain

$$\begin{aligned} \pi_1(-CA^{-1} + 1 + R + R^2 + \dots) &= F \\ \pi_1(-CA^{-1} + \frac{1}{1-R}) &= F \\ \pi_1 &= F(-CA^{-1} + \frac{1}{1-R})^{-1} \end{aligned} \quad (4.13)$$

where  $F$  is the vector of binomial probability distribution from the first equation of Eq. (4.6). Since we found  $\pi_1$ ,  $\pi$  follows using Eq. (4.9) and (4.10). We define  $P = [p_0, p_1, p_2, \dots]$  as the equilibrium probability distribution of queue length. It is easy to obtain  $P$  by

$$p_i = \sum_{j=0}^M \pi_{ij} \quad (4.14)$$

Analysis results will be presented and discussed in section 4.4.2.

### 4.3 Models for Scene-Cut Sequences

In this section, we obtain three models for Sequence 2. The first two are autoregressive

and Markov chain models. Since the assumptions of the birth-death model are violated by Sequence 2, it is replaced with another model, namely the *hidden Markov model*.

#### 4.3.1 Another Autoregressive and Markov Chain Model

Repeating the same procedure as in Section 4.2.1 and 4.2.2, we obtain an autoregressive model with order 2 as

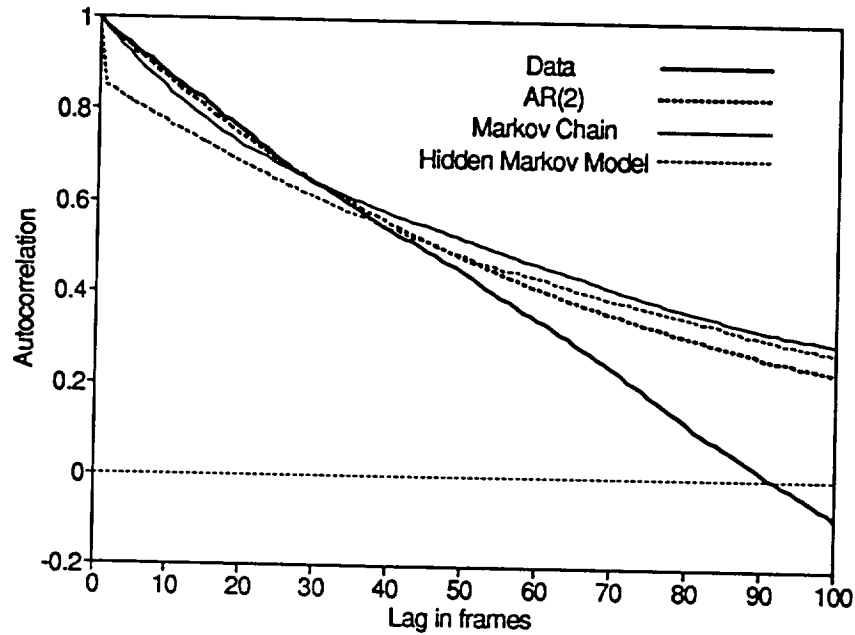
$$X_n = 1.291 + 0.814X_{n-1} + 0.169X_{n-2} + Z_n, \quad Z_n \sim WN(0, 0.018910) \quad (4.15)$$

It seems an AR(2) model is good enough to describe Sequence 2, both in terms of matching the autocorrelation function and in terms of the randomness of the residuals. The Markov chain model used here is the same as in section 4.2.2 except that a coarser quantizer and a larger number of states were used to cover the wide range of Sequence 2. The quantization stepsize is 0.1 bits/pixel and there are 29 states in our simulation. Figure 4.10 shows that the autocorrelation functions of these two models are reasonably close to that of Sequence 2, at least up to 40 frames.

#### 4.3.2 Hidden Markov Model (HMM)

The hidden Markov model (HMM) is a doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations [27]. An HMM is characterized by the following:

1.  $N$ , the number of states in the model.



**Figure 4.10** Autocorrelation function of several models (Sequence 2).

2.  $M$ , the number of distinct observation symbols per state.
3.  $A = \{a_{ij}\}$ , the state transition probability.
4.  $B = \{b_j(k)\}$ , the observation symbol probability.
5.  $\pi = \{\pi_i\}$ , the initial state distribution.

A HMM can be obtained as follows

1. Given the observation sequence  $O = O_1 O_2 \dots O_T$  and a model  $\lambda = (A, B, \pi)$ , compute  $P(O|\lambda)$ , the probability of the observation sequence, given the model.
2. Given the observation sequence  $O$  and the model  $\lambda$ , choose a corresponding state sequence  $Q = q_1 q_2 \dots q_T$ , which is optimal in some meaningful sense (i.e., best “explains” the observations).
3. Adjust the model parameters  $\lambda = (A, B, \pi)$  to optimize  $P(O|\lambda)$ .



Since Sequence 2 is generated from two different types of video, attaching the frame's content to the state of the HMM seems to be a natural choice. Two states are defined in the HMM model, namely the low-motion and the high-motion states, therefore  $N$  equals 2. We hope this choice can accurately reflect the bit rate distribution in different motion sequences. The observation symbols are the quantized bit-rate output of Sequence 2. The number of distinct observation symbols  $M$  is set to be 29 using the same quantizer as in Markov chain model. There are several possible ways of initializing the algorithm used for developing the HMM. These depend on the selection of matrix  $A$ ,  $B$ ,  $\pi$ . As suggested in [27], uniform estimates for the  $\pi$  and  $A$  parameters are adopted. As for the  $B$  matrix, weighted parameters are assigned in order to obtain the global maximum of the likelihood function in the algorithm.

#### 4.4 Goodness-of-Fit Tests

	Sequence 1		Sequence 2	
	Mean	Variance	Mean	Variance
Data	0.8296	0.0383	1.2905	0.5014
AR	$0.8292 \pm 0.0134$	$0.0392 \pm 0.0027$	$1.3251 \pm 0.0657$	$0.4303 \pm 0.0444$
MC	$0.8274 \pm 0.0090$	$0.0385 \pm 0.0012$	$1.3035 \pm 0.0613$	$0.4763 \pm 0.0324$
HMM	-----	-----	$1.2825 \pm 0.0621$	$0.4869 \pm 0.0326$

Table 4.1 Statistics with 95% confidence interval.

In this section some statistics, simulation and queueing analysis results are used to test goodness-of-fit for the models introduced above.

#### 4.4.1 Statistics

For each statistical model we generate 10 realizations of 5,000 frames. The 10 realizations are treated as independent and identically distributed samples. Table 4.1 shows mean, variance along with 95% confidence interval which is expressed as [34]

$$P[\bar{x} - t_{0.025}(\frac{s}{\sqrt{n}}) \leq x \leq \bar{x} + t_{0.025}(\frac{s}{\sqrt{n}})] = 0.95 \quad (4.16)$$

where  $s$  is sample's standard deviation,  $n$  is the number of samples which is 10. The value  $t_{0.025}$  equals 2.262 corresponding to 9 degrees of freedom for *Student's t* distribution. Note that it is possible for the AR model to generate negative values which are not allowed in our application. Since this situation does not occur often, negative values are simply replaced with zero and the statistics do not change drastically. Table 4.1 reveals that all confidence intervals include the values from the real data except the variance when applying AR(2) model in Sequence 2. Statistical test of hypothesis may suggest rejection of the AR(2) model at this point. Another powerful tool to test goodness-of-fit is *percentile* plot which draws the percentile of distribution. As shown in Figure 4.11, the Markov chain model shows a very good fit for the data from Sequence 1. The AR(4) model fits the data reasonably well except for some regions with small deviations. Figure 4.12 again shows that the AR(2) model is not a good model for sequence with clumps of large value, like Sequence 2. On the other hand, the Markov chain model and the HMM

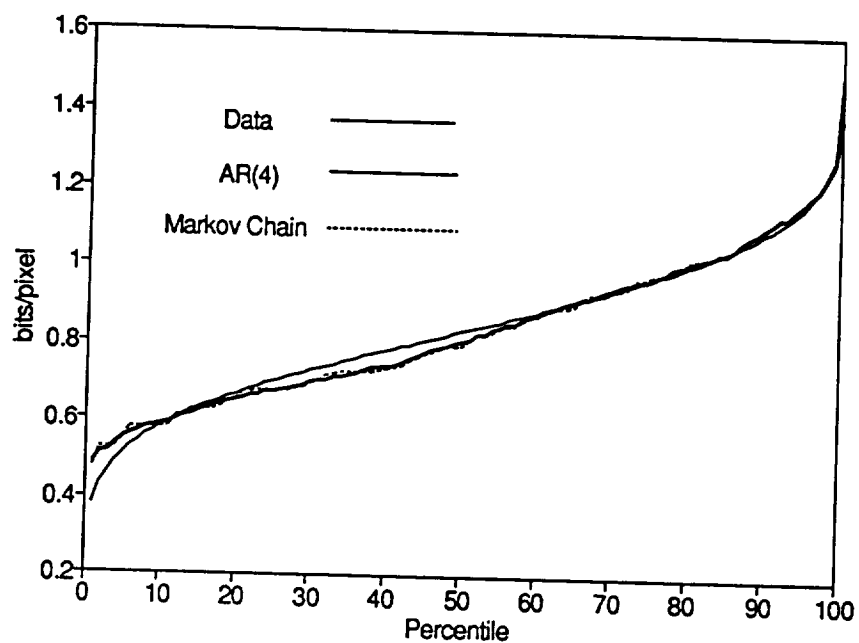


Figure 4.11 Percentile plot of several models (Sequence 1).

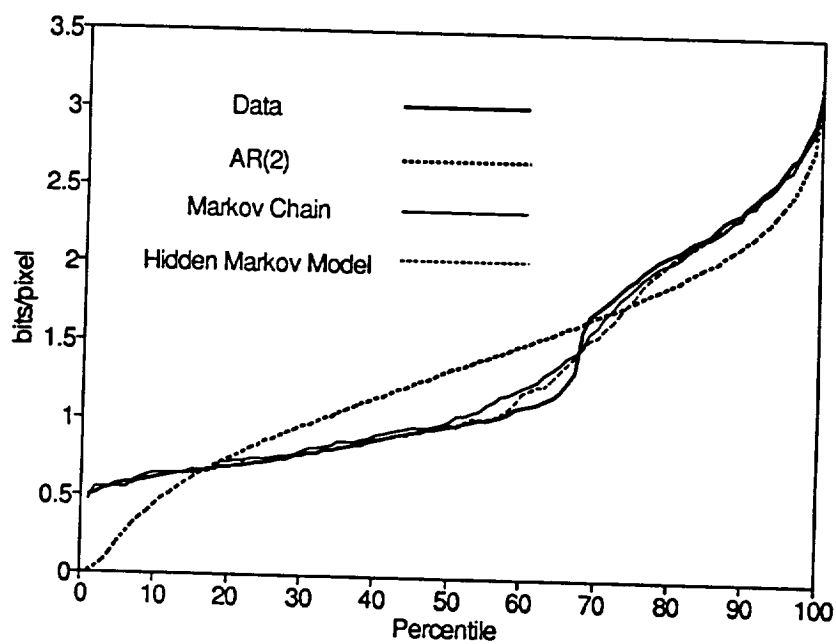


Figure 4.12 Percentile plot of several models (Sequence 2).

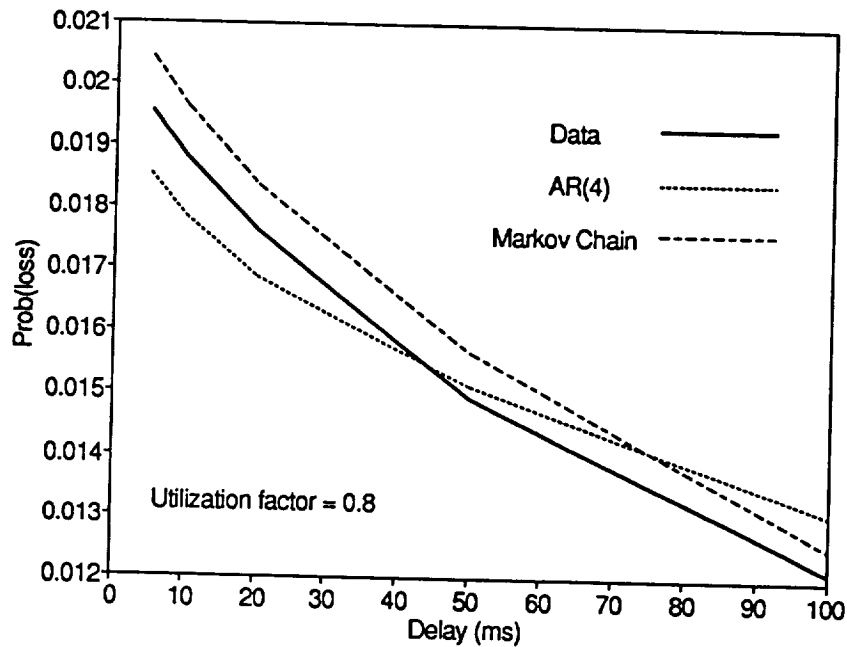
satisfactorily follow the big jump of data.

#### 4.4.2 Cell Loss Performance for Homogeneous Sequence

Cell loss is one of the most important problems when transmitting information through ATM networks. Therefore, besides fitness of statistics, queueing performance is another important consideration in video source modelling [32]. We design two queueing systems to perform the evaluation. The first one is a finite buffer queue where buffer space corresponds to transmission delay. Every arriving cell is discarded when the buffer is full. The second one is a queue with infinite buffer space. The reason this queueing model is proposed is to compare the performance of minisource model with other models. All coding bits from a single frame are assembled into cells with length equals 384 bits, according to the ATM cell format specified in Chapter 2. All the cells from the same frame are equally spaced in a frame duration, 1/30 second. The server serves with a determinant rate which equals the average video rate divided by a utilization factor.

	Cell Loss Probability for Various Maximum Delay Allowed				
	5ms	10ms	20ms	50ms	100ms
Data	0.0195	0.0188	0.0176	0.0149	0.0121
AR(4)	$0.0185 \pm 0.0032$	$0.0178 \pm 0.0032$	$0.0168 \pm 0.0032$	$0.0151 \pm 0.0033$	$0.0131 \pm 0.0033$
MC	$0.0204 \pm 0.0022$	$0.0196 \pm 0.0021$	$0.0183 \pm 0.0021$	$0.0156 \pm 0.0020$	$0.0125 \pm 0.0019$

Table 4.2 Cell loss probability of several models (Sequence 1, Case 1).



**Figure 4.13** Cell loss probability of several models (Sequence 1, Case 1).

#### **Case 1:** Queueing model with finite buffer

Figure 4.13 shows the cell loss probability of Sequence 1, AR(4) and Markov chain model from simulation with utilization factor  $\rho = 0.8$ . With the determinant transmission rate, 1 ms delay is approximately equivalent to the transmission time of one cell. Therefore, the cell loss probability corresponding to, say 10 ms delay, is the simulation result with a 10 cell buffer. For the AR(4) and Markov chain models, simulations are run with 10 realizations each. The values shown in Figure 4.13 are the average of 10 simulations. The difference between the performances of data and models seems small. However, we would like to validate the fitness of these models by checking the test of hypotheses again using  $t$  statistics. Table 4.2 shows that both models easily pass the test.

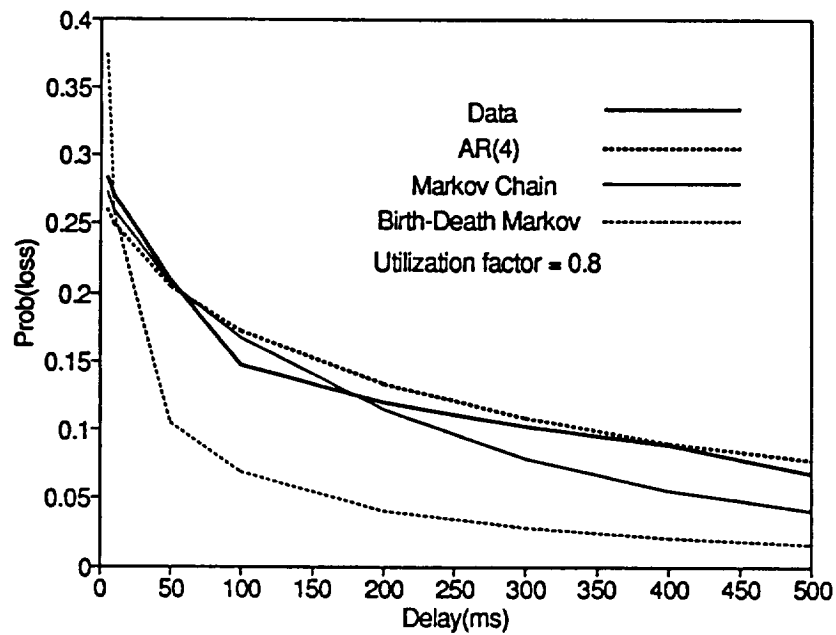
**Case 2: Queueing model with infinite buffer**

This queueing model is similar to the previous one but with an infinite buffer space. Therefore all arriving cells are put in the buffer without loss. We collect time average statistics to find the distribution of buffer length. Probability of cell loss in this case is equivalent to the probability that queue length exceeds some certain value  $k$ . It means that the arriving cell who finds there are more than  $k$  cells already waiting in the queue is deemed to be lost because it will be too late for video reconstruction at the receiver end. As in the previous case, simulations have been run using Sequence 1 and 10 realizations from the AR(4) and Markov chain models each. Comparison between simulation results and the queueing analysis result of a minisource model will also be made.

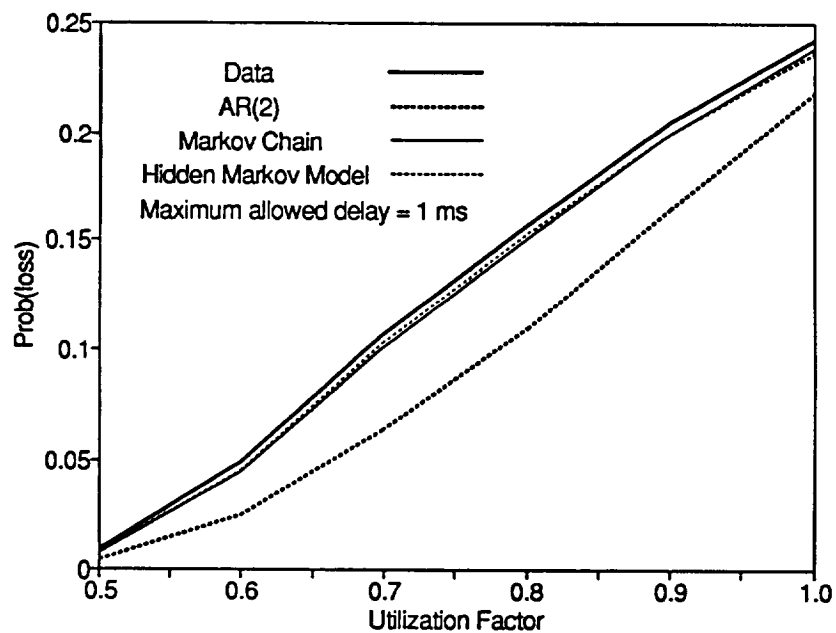
As described in Section 4.2.3, we were able to find the equilibrium distribution of queue length  $P$  in the minisource model. The parameters of the model  $M$ ,  $A$ ,  $\alpha$ ,  $\beta$  are obtained by matching Eq. (4.6) with measured values. Taking  $M$  as a parameter, we have

$$\begin{aligned}
 \beta &= a / \left(1 + \frac{E^2(\lambda)}{M \cdot C(0)}\right) \\
 &= 1.5 / \left(1 + \frac{0.6883}{M \cdot 0.0383}\right) \\
 \alpha &= a - \beta = 1.5 - \beta \\
 A &= \frac{C(0)}{E(\lambda)} + \frac{E(\lambda)}{M} = 0.0462 + \frac{0.8296}{M}
 \end{aligned} \tag{4.17}$$

In order to cover the value of maximum bit-rate,  $M$  is chosen to be 13. In this case,  $\alpha$  and  $\beta$  are 0.8699 and 0.6301 per second respectively and  $A$  is 0.11 bits/pixel. We define  $\lambda$  as the cell generating rate per second per minisource and  $\mu$  as the cell transmission rate per second. The utilization factor is set to be 0.8.



**Figure 4.14** Cell loss probability of several models (Sequence 1, Case 2).



**Figure 4.15** Cell loss probability of several models (Sequence 2, Case 1).

Figure 4.14 shows the simulation and analysis results. Note that the minisource model overestimates cell loss probability in the small delay region and underestimates that in the large delay region. The over-estimation is caused by the fact that Sequence 1 does not contain low bit-rate which consequently make  $\alpha$  large. A large value of  $\alpha$  means that if a minisource is on it tends to stay on. In the case of a short buffer, this kind of behavior causes cell loss. A long buffer can absorb this effect and reflect the “smooth” characteristic of a birth-death process. However, it is this “smooth” property that causes the underestimation in the long delay area. From the observation of Sequence 1, there are some transitions between states which are not neighboring. Nevertheless, the minisource model is still a powerful analytic tool, especially when investigating the behavior of statistical multiplexing.

#### 4.4.3 Cell Loss Performance for Scene-Cut Sequence

Cell Loss Probability for Various Utilization Factor					
	1.0	0.9	0.8	0.7	0.6
Data	0.2424	0.2046	0.1577	0.1067	0.0491
AR(2)	$0.2176 \pm 0.0205$	$0.1642 \pm 0.0187$	$0.1100 \pm 0.0159$	$0.0635 \pm 0.0127$	$0.0245 \pm 0.0085$
MC	$0.2383 \pm 0.0195$	$0.1995 \pm 0.0166$	$0.1516 \pm 0.0133$	$0.1006 \pm 0.0101$	$0.0442 \pm 0.0060$
HMM	$0.2360 \pm 0.0210$	$0.1992 \pm 0.0177$	$0.1535 \pm 0.0135$	$0.1029 \pm 0.0087$	$0.0451 \pm 0.0035$

Table 4.3 Cell loss probability of several models (Sequence 2).



From Figure 4.2 we observed the durations which contain low and high bit-rate value alternatively. Apparently, the long periods of high bit-rate data will dominate the probability of cell loss, if transmission rate is less than the average of the high bit-rate period. The Markov chain model has the ability to generate a collection of large frames because it follows the state transition probability. The hidden Markov model also can produce clumps of large frames since it has a “hidden” high bit-rate state. In this simulation, the utilization factor is a variable and the delay constraint is set to be 1 ms. It is shown clearly, from Figure 4.15, that Markov chain model and the HMM closely follow the performance of recorded data but that the AR(2) model fails to do that. The AR(2) model greatly underestimates the probability of cell loss since it was not able to produce consecutive large frames. Table 4.3 shows the simulation along with *t*-test results.

## 4.5 Some Notes

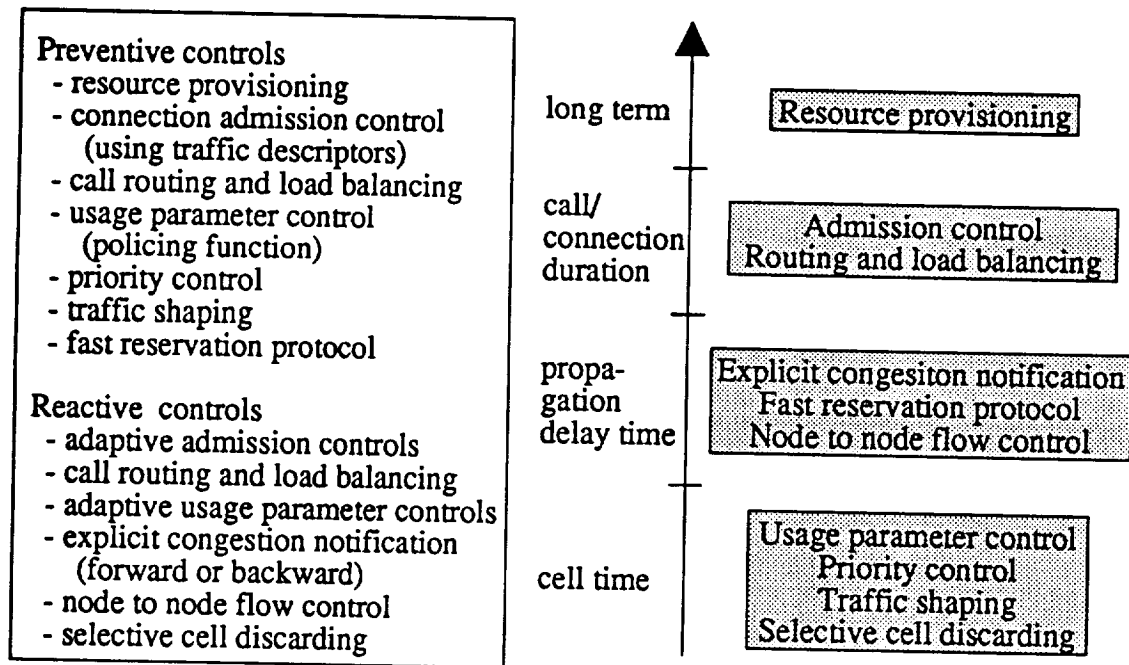
It has been shown that the AR model with moderate order can properly model homogeneous video sources. Markov chain models have excellent simulation performance for both sequences but without analysis significance. The minisource model is basically a two-phase burst/silence model, which is used extensively in speech source modelling. Aggregate minisource models with birth-death property can model smooth processes and are analytically traceable. In the study of statistical multiplexing, an aggregate minisource is an accurate model because a multiplexed source is smoothed out according to the rule of large numbers. HMM is believed to be a good simulation model since it can handle complicated sequences with underlying stochastic processes. It could be realistic for a

long video transmission. Finally, it is noted that the effect of packetization is not ignored in the simulation since we did not adopt fluid flow approximation. Congestion control for video transmission in ATM networks will be discussed in the next chapter, simulations will be run with models which are justified in this chapter.

## Chapter 5

# Network Congestion Control

Congestion is defined as a state of network elements - switches, concentrators, transmission links - in which, due to traffic overload and/or control resource overload, the network is not able to guarantee the QOS of established connections and/or acceptance of new connection requests. The promising integration technique of ATM networks raises many new congestion control problems due to the higher degree of resource sharing compared to the conventional synchronous transfer mode (STM) networks. Conventional congestion control methods are basically *reactive controls* which take necessary actions to recover from a congested situation [36]. Such mechanisms, like window-based flow control schemes, typically rely on the end-to-end exchange of control messages in order to regulate traffic flow. However, the high transmission speed of ATM networks make it difficult to perform any type of reactive congestion control because the propagation delays across the network usually dominate switching and buffering delays. Also, the nature of traffic in future ATM networks is significantly different which affects the design of the congestion control. Unlike traditional circuit-switching networks where the



**Figure 5.1** Classification of ATM congestion control mechanisms [39].

bandwidth requirements are fixed, ATM networks will face emerging services which have an intrinsic rate, possibly varying by several orders of magnitude, determined by external factors outside the control of the network. Meanwhile, real-time traffic (e.g., voice, video and image) will require some level of bandwidth guarantee. The different scenario drives the investigation of many new concepts and approaches [35]. Different from reactive controls, *preventive controls* take actions to prevent congestion from occurring. Figure 5.1 shows the various control mechanisms proposed for ATM networks. Although a significant amount of research has been done in this area, congestion control problems are still considered to be difficult. It is believed [37] that a satisfactory network performance relies on simultaneous application of multiple congestion schemes whose actions are related to the various levels of activity - connection, burst, and cell level.

In this chapter, several interesting proposals of congestion control which have significant effect or useful application in video transmission will be introduced. Particularly, we try to address the network congestion problem solely from the view point of the user, e.g., a video codec. Interactions between video codec and possible congestion controls will also be investigated.

## 5.1 Call Admission Control

At the connection level, the connection admission control decides whether a new call could be admitted under the requested traffic and QOS parameters. For constant bit-rate services and other connections that require peak bandwidth allocation, the decision to accept a new connection is relatively straightforward. This type of connection should be accepted only if its peak bandwidth is available at each node along the connection's selected path. For variable bit-rate connection, however, the connection cannot be accepted only on the basis of the available network resources; its effect on the QOS of all other existing connections sharing the same resources with the new connection should also be taken into account. In the initial negotiation phase, the user sends a signaling message to the network typically containing peak and average bit rates and some measure of the frequency and length of peak periods. The QOS parameters also have to be included. They may be expressed as average cell transfer delay, delay jitter and cell loss probability.

In other words, the call admission control is a resource allocation scheme which maintains the balance between QOS and network utilization by controlling the number

of calls in the network. In order to do this, the network should know both the current traffic and the traffic characteristic of the new call before it can decide whether to accept or reject the new call. Basically, call admission control is a challenge that the network specialists must face. They have to propose a set of traffic parameters which should be convenient for the user to specify and easy for a network provider to monitor. In addition, it should involve enough information to estimate network performance after acceptance of a new connection. Candidates for traffic parameters are: peak-bit-rate (PBR), utilization and average burst length; PBR, average-bit-rate (ABR) and burstiness ( $= \text{PBR}/\text{ABR}$ ); PBR, ABR and bit rate variance; maximum call throughput in short and medium duration; and PBR, ABR, and average peak duration. Many call admission control schemes have also been proposed in the literature [45,46,47,48,49,50,51,52,72]. They differ mainly in the approaches of characterizing traffic and predicting future traffic.

An idea called “equivalent bandwidth” has been proposed in [46]. This bandwidth represents the equivalent amount of link capacity that is needed for a particular connection. It is a function of both the characteristics of individual connections and their interaction within a link. The desired network QOS is met only if, at all links, the aggregate equivalent bandwidth of connections remains below link capacity. The estimation of equivalent bandwidth is not an easy task because it must account not only for individual connection characteristics but also for interactions with concurrent transmissions in the link. In addition, it must be performed “on-line,” i.e., track changes in link loads as connections are added or removed from the network.

In [56], a connection is characterized by a call metric vector  $(R_{peak}, \rho, b)$ , where  $R_{peak}$

is the peak rate at which the source can generate data,  $\rho$  is the utilization or fraction of time it is active and source transmitting at  $R_{peak}$  and  $b$  is the average burst duration. Using this metric, the equivalent bandwidth required for a new connection can be calculated with the combination of two approximations. The first one is the fluid-flow approximation and focuses on a connection in isolation. The second one is the stationary approximation which gives the bandwidth requirements and is suggested for use when the effect of statistical multiplexing is more important. Both approximations overestimate the actual value of the equivalent bandwidth and are inaccurate for different ranges of connection characteristics. Therefore, the final estimation is the minimum of both approximations.

The bandwidth  $\hat{c}$  required by an individual connection with metric vector  $(R_{peak}, \rho, b)$  can be estimated using a simple fluid-flow model, and is given by

$$\hat{c} = \frac{\alpha b(1-\rho)R_{peak} - x + \sqrt{[\alpha b(1-\rho)R_{peak} - x]^2 + 4x\alpha b\rho(1-\rho)R_{peak}}}{2\alpha b(1-\rho)} \quad (5.1)$$

where  $x$  represents the available buffer space, and  $\alpha = \ln(1/\epsilon)$  with  $\epsilon$  the desired loss probability. On the other hand, the amount of bandwidth required by  $N$  connections multiplexed on the same link can be approximated by

$$\hat{B} = m + \gamma\sigma, \quad \text{with } \gamma = \sqrt{-2\ln(\epsilon) - \ln(2\pi)} \quad (5.2)$$

where  $m$  is the mean aggregate bit rate ( $m = \sum_{i=1}^N m_i$ ) and  $\sigma$  is the standard deviation of the aggregate bit rate ( $\sigma^2 = \sum_{i=1}^N \sigma_i^2$ ). The final estimate of the amount of link capacity  $\hat{C}$  required by a set of  $N$  connections is then obtained from a simple combination of the above approximations

$$\hat{C} = \min\left\{m + \gamma\sigma, \sum_{i=1}^N \hat{c}_i\right\} \quad (5.3)$$

It has also been mentioned that, based on above estimation, it is possible to define *link metrics* which characterize the current capacity allocation on network links [46]. This metric provides a simple and compact form to store the bandwidth allocation information, while allowing for real-time updates and computations of allocation levels. A three-dimensional vector representation meeting these requirements is provided as

$$L_j = \left( m = \sum_{i=1}^N m_i, \sigma^2 = \sum_{i=1}^N \sigma_i^2, \hat{C}_{(N)} = \sum_{i=1}^N \hat{c}_i \right) \quad (5.4)$$

where  $N$  is the number of connections currently multiplexed on link  $j$ ,  $m$  and  $\sigma^2$  are the mean and variance of the aggregate bit rate, and  $\hat{C}_{(N)}$  is the sum of the  $N$  individual equivalent connection bandwidths computed from Eq. (5.1). An important feature of Eq. (5.4) is that it allows for incremental updates of link metric vectors as connections are added or removed. Specifically, a request for connection establishment or removal with call metric vector  $(R_{peak}, \rho, b)$  is used to compute a connection request vector  $r$  of the form

$$r = (m, \sigma^2, \hat{c}) \quad (5.5)$$

The new link metric vector after adding (removing) a connection with request vector  $r$  is simply given by the component-wise addition (subtraction) of  $L_j$  and  $r$ .

For a single transmission without statistical multiplexing, the equivalent bandwidth is equal to the transmission rate which satisfies the QOS requirement. Generally, the equivalent bandwidth is between the peak bandwidth and the mean bandwidth. In ATM networks, video transmission may require per-connection QOS but allow for statistical



multiplexing. Taking advantage of multiplexing gain, it is possible to assign a lower equivalent bandwidth to a video transmission depending on how many connections belonging to the same class are in the same link. It is also interesting to know if it is possible to reduce the equivalent bandwidth requirement of a video transmission by changing traffic characteristics, either through shaping or coding.

The approximation method described above uses a two-state fluid-flow model to capture the basic behavior of the source through a metric vector  $(R_{peak}, \rho, b)$ . Such a source is either in an "idle state," transmitting at zero bit rate, or in a "burst state" and transmitting at its peak rate. However, a variable bit rate video source is more like a multiple state model with each state representing a quantization level. As validated in Chapter 4, the Markov chain model is a sufficient model for both kinds of video sequences. It is noted that our Markov chain model does not take into account the variation of the bit rate within a single frame. The cells are sent equidistantly according to the actual bit rate of the frame; this represents the smoothest flow. The cells can also be sent equidistantly according to the maximum bit rate of the codec, which implies a pattern with one burst and one silence period during each frame. This is the most bursty case and can be used to approximate equivalent bandwidth.

Extensive simulations were performed to estimate equivalent bandwidth for different sequences. QOS is expressed as delay ( $t_d$ ) and cell loss probability ( $p$ ). In the simulations,  $p$  is set to be  $10^{-5}$  for illustration, some services may require lower  $p$ ;  $t_d$  is on the order of milliseconds. The total number of cells generated to estimate a loss probability of  $10^{-5}$  was about  $10^8$ - $10^9$ . In the simulations, delay represents the time for sending cells of one

full buffer. In Figures 5.2 and 5.3, 1 *ms* delay equals the time for transmitting approximately 5.51 and 8.28 cells respectively. In another word, simulation result for 5 *ms* in Figure 5.2 is conducted with buffer space of 27 cells. Figure 5.2 shows the equivalent bandwidth assignment versus the delay requirement for one homogeneous video transmission. The curve in Figure 5.2 was obtained by means of an inversion process from the cell loss probability versus delay. The two straight lines of Figure 5.2 represent the bandwidth necessary in case of peak rate and average rate bandwidth allocation. Figure 5.3 displays the result for the video sequence with scene cut. It is observed that the equivalent bandwidth is very close to peak bandwidth. Thus, peak rate allocation might be suggested for such a single transmission. Figure 5.4 shows the variation of equivalent bandwidth as a function of the number of sources multiplexed. Ten sequences are generated from the same Markov chain model using different seeds. Using network simulation, the average of equivalent bandwidth for these ten sequences is obtained with  $p_l$  and  $t_d$  as defined. The concavity of equivalent bandwidth exhibits the gain obtained by statistical multiplexing. The curve, namely approximate bandwidth, is obtained from the approximation method. As mentioned above, cells can be generated as one burst and one silence period during each frame. Therefore, a metric vector  $(R_{peak}, \rho, b)$  of the video source can be declared for such burst/silence pattern with  $\rho$  expressed as  $R_{mean}/R_{peak} = 2.11/3.48 = 0.605$ , and  $b = \rho/30 = 0.02$  second. This overestimated approximation combined with a most bursty traffic description should give an upper bound for equivalent bandwidth. From Figure 5.4, we can see that the approximate bandwidth provides a reasonably good estimation of the equivalent bandwidth from

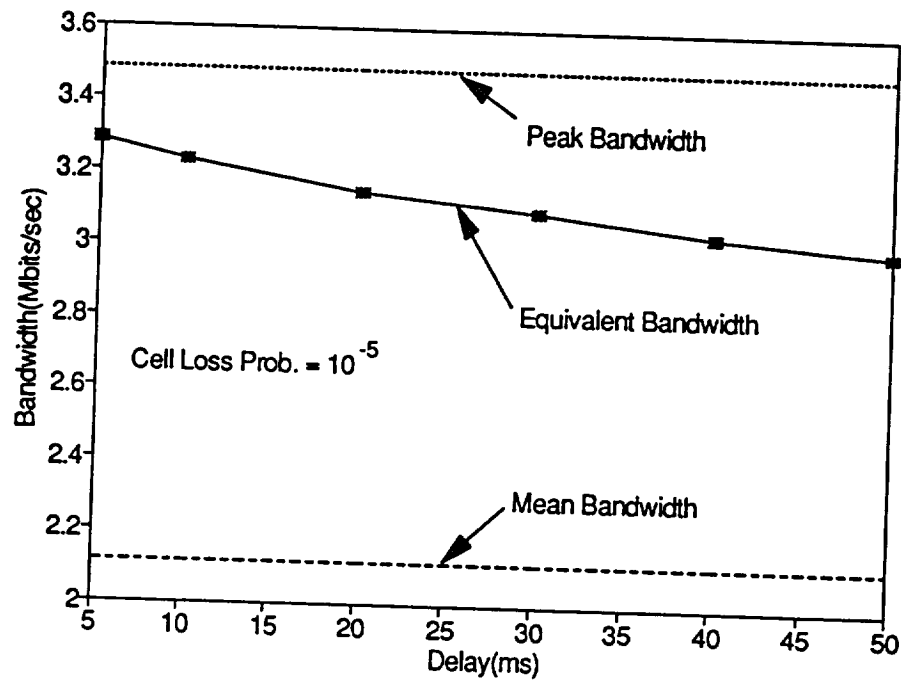


Figure 5.2 Influence of delay on equivalent bandwidth for one homogeneous sequence.

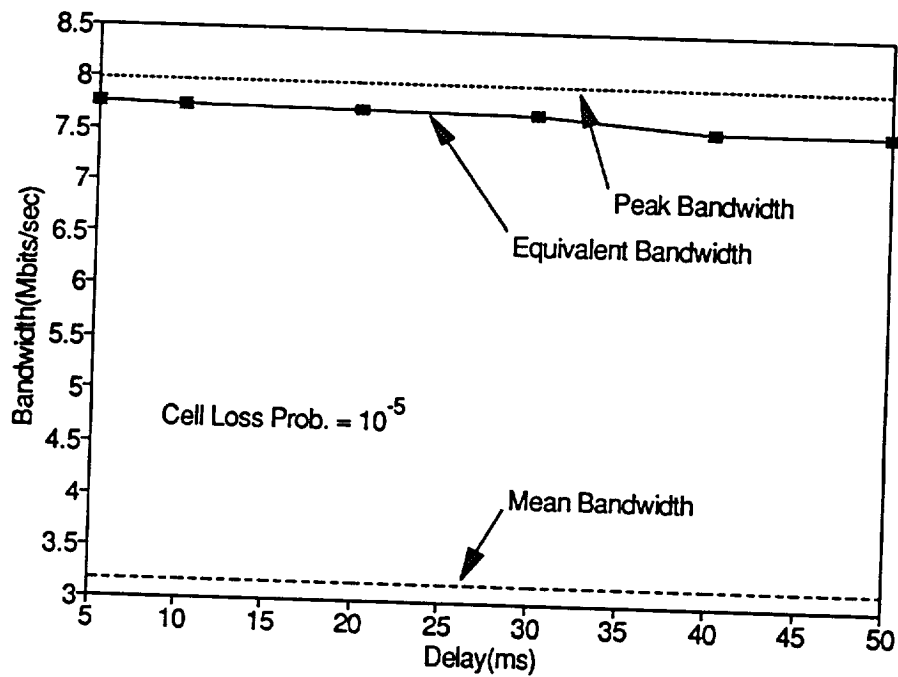
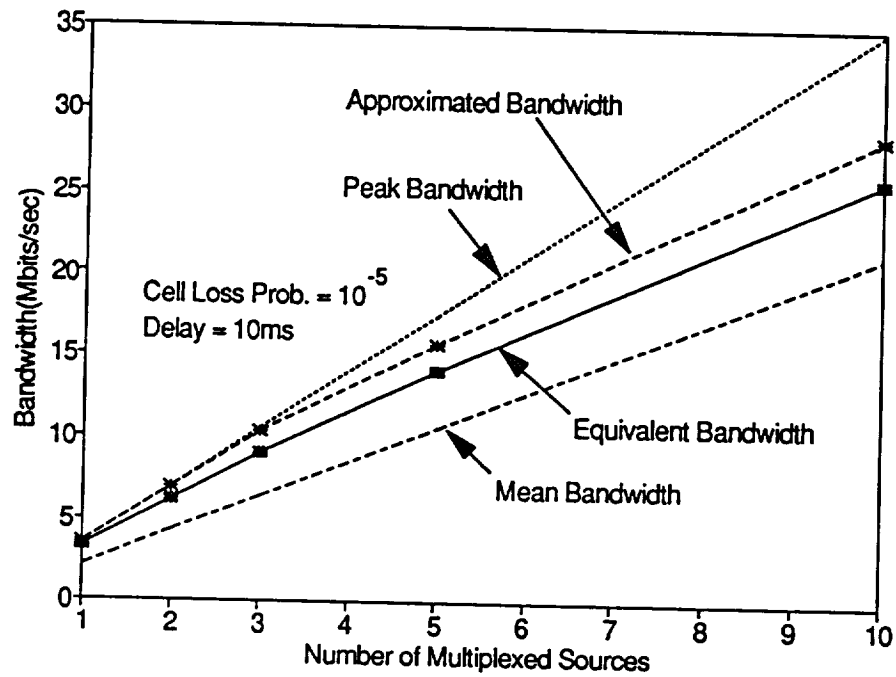
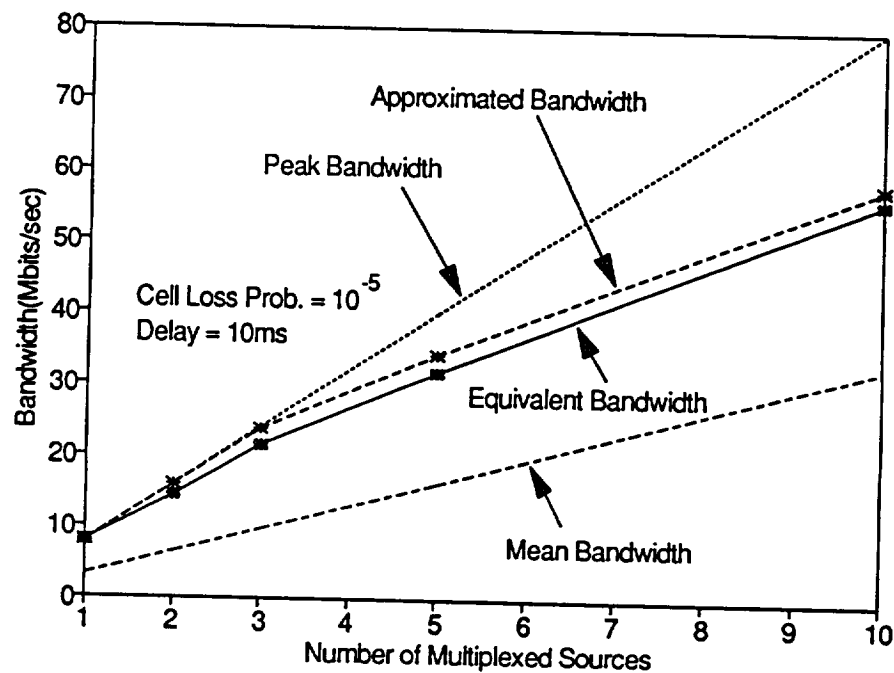


Figure 5.3 Influence of delay on equivalent bandwidth for one scene-cut sequence.



**Figure 5.4** Equivalent and approximated bandwidth for various number of multiplexed sources (homogeneous sequence).



**Figure 5.5** Equivalent and approximated bandwidth for various number of multiplexed sources (sequence with scene-cut).

simulation. Figure 5.5 shows that a good multiplexing gain can also be expected for the scene-cut sequence. Considering actual intraframe behavior, the true equivalent bandwidth will be somewhere between the approximate and equivalent bandwidth. It is believed that the equivalent bandwidth in Figure 5.4 and 5.5 can be realized by providing adequate buffering in the codec. In the following, equivalent bandwidth assignment will be adopted as our call admission control scheme.

## 5.2 Usage Parameter Control

Usage parameter control (UPC), which is a function that supervises the connection, has been given many other names such as *policing function*, *flow enforcement*, *traffic enforcement*. The purpose of the UPC is to ensure that, during the information transfer phase, the traffic stream arriving from each connection conforms to its agreed contract. In other words, the UPC must ensure that excess traffic from one connection does not in any way undermine the QOS of other established connections. Although no specific monitoring control method has been standardized, some desirable features have been identified in Recommendation I.311 [57]:

- *Capability of detecting any traffic situation which may cause QOS degradation.*
- *Selectivity over the range of checked parameters.*
- *Speed in perceiving the incorrect behavior.*
- *Efficient answer to contract violations in order to prevent traffic overloads.*
- *Simplicity of implementation.*

There has been considerable discussion regarding which parameters should be monitored. Candidates include peak bandwidth, average bandwidth, burst length, etc.. CCITT Recommendation I.311 suggests the same parameters already referred for source characterization in the connection admission procedure as possible policing parameters. When monitoring the peak cell rate, a certain amount of tolerance must be included to account for cell delay variation and jitter. For connections that need per-connection QOS but do not require peak bandwidth allocation, fair monitoring is much more complex than peak bandwidth allocated connections.

The actions on the violating cells are important not only to solve congested situations but also to discourage violations. Three possible actions, with different efficiencies, seem possible:

- ***Discarding all violating cells:*** only the agreed cells get into the network. It has been shown that an unrealistically huge bucket is necessary to guarantee reasonable cell loss probability under this discipline [58]. It is possible to increase the leaking rate to maintain a small bucket, but unfortunately this approach also decreases the ability of detecting violations.
- ***Charging or breaking the connection:*** requiring an accurate policing mechanism since this action is somewhat drastic.
- ***Marking violating cells:*** by marking violating cells, the network can treat them with lower priority in the rare event that selective cell discarding is required. Some people argue that this approach may not be efficient during congestion periods [37].

Another question about monitoring is its location. To be able to protect all network resources, the policing function must be located as close as possible to the source. Of course it must still remain under the direct control of the network provider. Depending on the customer access network configuration, the policing function may be performed on VC's, on VP's, or on the total traffic volume on an access link within components like concentrators, local exchanges, and ATM cross-connects.

Various UPC mechanisms have been suggested for ATM networks, including [58]:

- *Leaky Bucket (LB)*

The LB mechanism consists of a counter which is incremented by one when a cell arrives and decremented by one for fixed time as long as the counter value is positive. When the cell arrival rate exceeds the decrementation rate, the counter value starts to increase. It is assumed that the source has exceeded the admissible parameter range if the counter reaches a predefined limit, and suitable actions are taken on all subsequently arriving cells until the counter has fallen below its limit again. It has been proposed that the  $G/D/1$  delay loss system [60] is an exact model for the violation probability of the LB mechanism, which is identical to the cell loss probability if violating cells are discarded.

- *Jumping Window (JW)*

The JW mechanism limits the maximum number of cells accepted from a source within a fixed time interval (*window*) to a number  $N$ . The new interval immediately follows the previous interval (*jumping window*) and the counter value is restarted again with an initial value of zero. The probability that policing actions must be taken on a cell

can be computed by using the counting process for the cell arrivals, which characterizes the number of arriving cells in an arbitrary time interval [58].

- *Triggered Jumping Window (TJW)*

The time window of the JW mechanism is not synchronized with source activity. To avoid the ambiguity problems arising from the fact, the TJW mechanism has been proposed, where the time windows are not consecutive but are triggered by the first arriving cell. The TJW mechanism can be analyzed in a similar way as the JW mechanism with little modification.

- *Exponentially Weighted Moving Average (EWMA)*

The EWMA mechanism uses fixed consecutive-time windows like the JW mechanism. The difference is that the maximum number of accepted cells in the  $i$ th window ( $N_i$ ) is a function of the allowed mean number of cells per interval  $N$  and an exponentially weighted sum of the number of accepted cells in the preceding intervals ( $X_j$ ) according to the rule

$$N_i = \frac{N - \gamma S_{i-1}}{1 - \gamma} \quad 0 \leq \gamma < 1 \quad (5.6)$$

with

$$S_{i-1} = (1 - \gamma)X_{i-1} + \gamma S_{i-2} \quad (5.7)$$

If  $\gamma = 0$ ,  $N_i$  is a constant and the mechanism is identical to JW algorithm. The EWMA mechanism can be evaluated by using event-by-event simulation.

- *Moving Window (MW)*



Similar to the JW mechanism, the maximum number of cell arrivals within a given time interval  $T$  is limited by this mechanism. The difference is that each cell is remembered for exactly one window width. That is, the arrival time of each cell is stored and counter is incremented by one for each arrival. Exactly  $T$  time units after the arrival of an accepted cell the counter is decremented by one again. The MW mechanism can be modeled by a multiple server loss system, where the deterministic service time reflects the window width  $T$  and the number of servers is defined by the maximum allowed number  $N$  of cells in the interval.

The long-term average cell rate can be determined by the maximum accepted number of cells per interval  $N$ , the window width  $T$  and the decrementation interval  $D$  as

$$\begin{aligned} T &= \frac{N}{\lambda C} = \frac{N}{\lambda_p} & \text{for window based mechanisms} \\ D &= \frac{1}{\lambda C} = \frac{1}{\lambda_p} & \text{for Leaky Bucket} \end{aligned} \quad (5.8)$$

where  $C$  is an overdimensioning factor introduced to decrease violation probability which usually is too large when mean cell rate policing is adopted. Figure 5.6 shows the behavior of the counter state for different mechanisms. In this example,  $N$  equals 2 and a mean cell rate  $\lambda = 1/2$  arrivals per time unit is assumed which result in  $T = 4$  time units and  $D = 2$  time units. The dotted line for EWMA mechanism represents the actual policing limit depending on  $\gamma$ . Under identical condition, the following inequality holds:

$$P_{viol,MW} \geq P_{viol,TJW} \geq P_{viol,JW} \geq P_{viol,EWMA} \quad (5.9)$$

A comparison and analysis using burst/silence source model for above UPC algorithms has been presented in [58]. In order to test their monitoring ability for video connections,

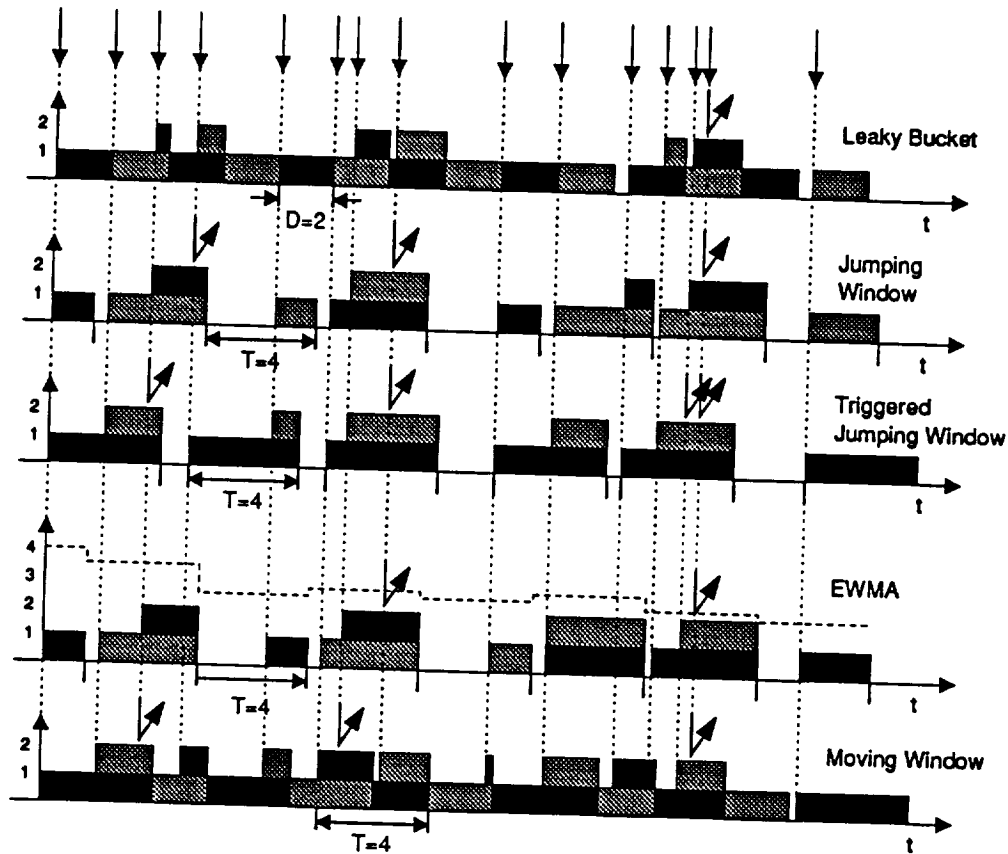
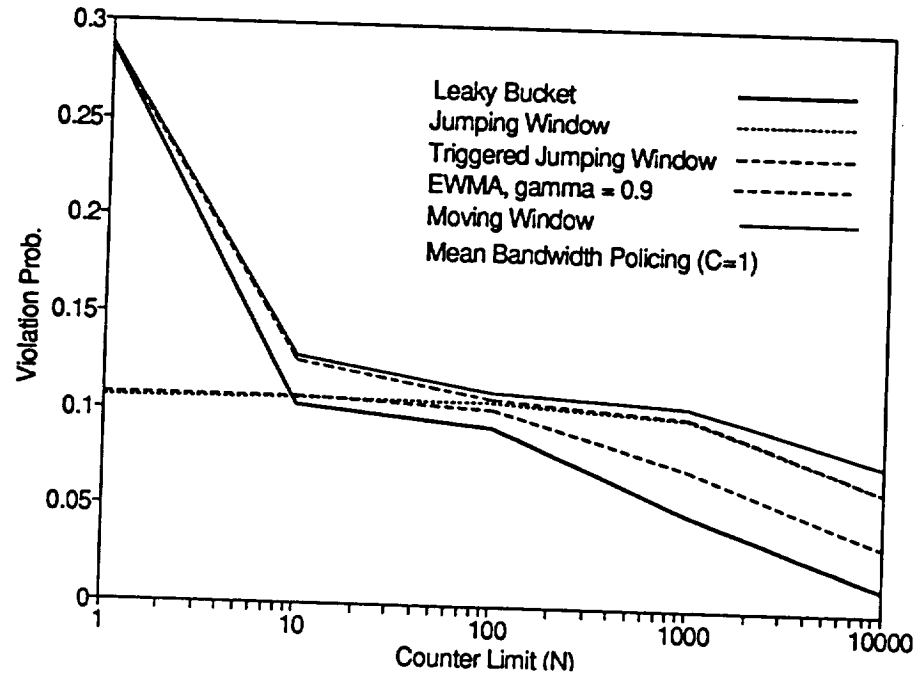


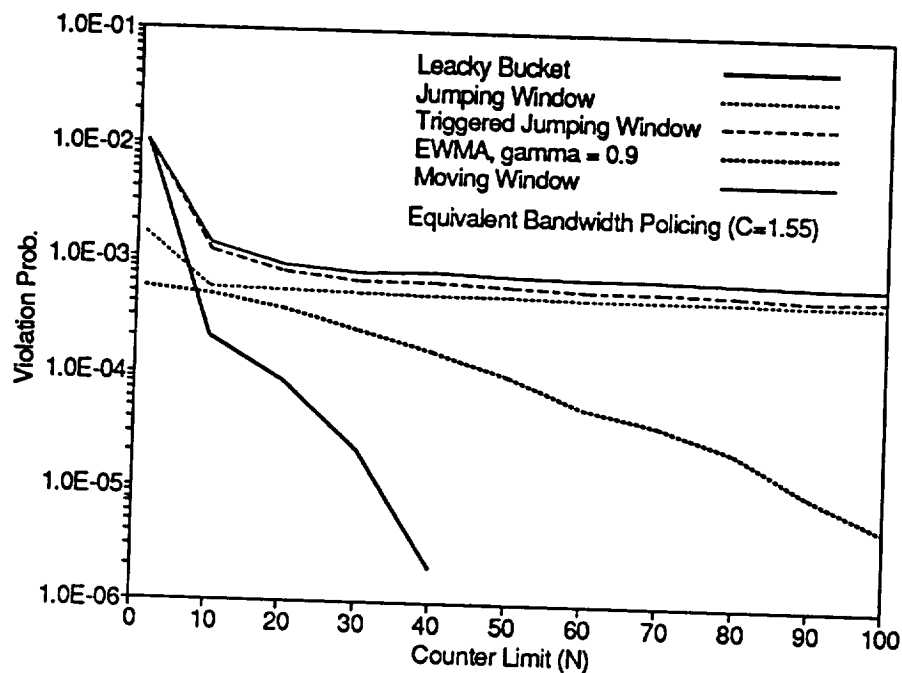
Figure 5.6 Example of counter state for different mechanisms [58].

simulations were carried out using a homogeneous video sequence for illustration. Figure 5.7 shows the influence of the counter limit on the violation probability. It is observed that the violation probability is not too sensitive to the counter limit when using mean cell rate policing. In fact none of above mechanisms can successfully enforce the source to its mean rate. In order to avoid excessive cell loss, an unrealistically huge counter is needed to monitor over a large window, which implies inefficient control of the traffic flow. If equivalent bandwidth is allocated when the connection is set up, then instead of

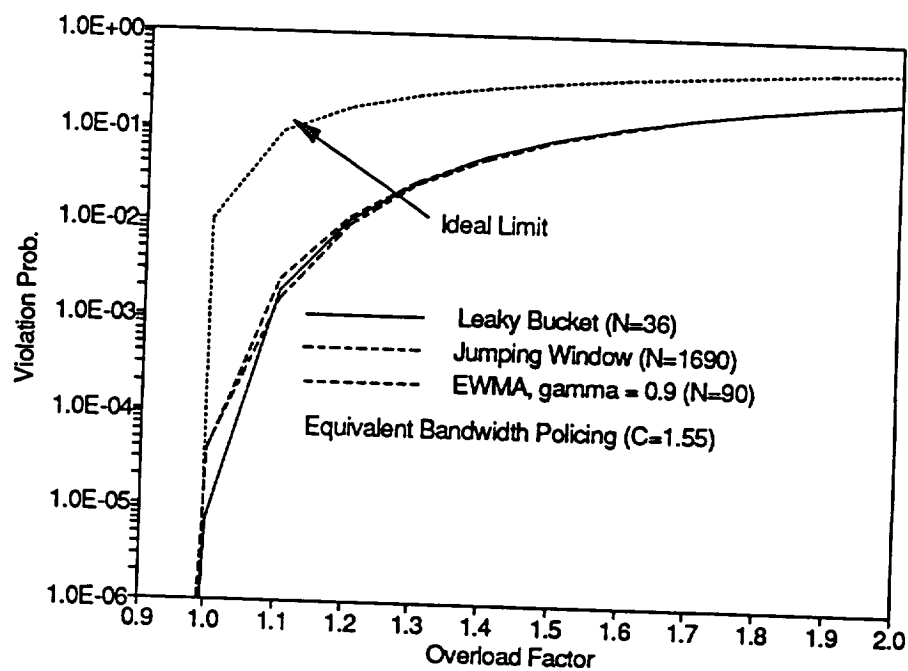


**Figure 5.7** Influence of counter limit on violation probability using mean bandwidth policing.

monitoring the mean cell rate, an equivalent bandwidth policing is more appropriate. Figure 5.8 shows that violation probability less than  $10^{-5}$  can be achieved by the LB with a counter limit  $N = 40$ . The EWMA mechanism also shows a satisfactory performance with  $\gamma = 0.9$ . The violation probability is still considered high for other window mechanisms with a counter limit up to 100. The detection ability of long-term overload for the LB, JW and EWMA algorithms is shown in Figure 5.9. To be able to dimension these mechanisms to a violation probability of about  $10^{-5}$ , different counter limits have been assigned as  $N_{LB} = 36$ ,  $N_{JW} = 1690$  and  $N_{EWMA} = 90$ . The ideal limit curve is calculated from the percentage over the mean cell rate and describes an ideal policing behavior. Despite the low counter limit, the LB mechanism is able to perform as well as the JW



**Figure 5.8** Influence of counter limit on violation probability using equivalent bandwidth policing.



**Figure 5.9** Overload detection ability of LB, JW and EWMA mechanisms using equivalent bandwidth policing.

and EWMA mechanisms. Figure 5.9 did not include TJW and MW, since both algorithms are less sensitive than the JW mechanism [58]. Although EWMA has comparable performance to the LB mechanism, practically it is more complicated to implement.

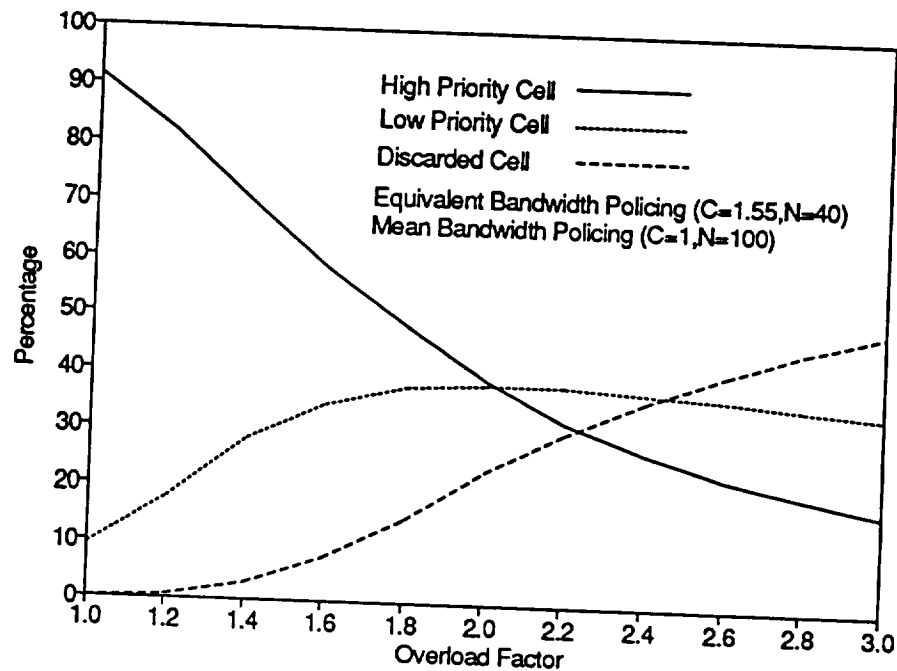
The optimum policing mechanism should have the flexibility to cope with short-term statistical fluctuations and the ability to detect long-term overloads. The effectiveness of the optimum mechanism, on the other hand, depends heavily on the characteristics of the traffic sources and their QOS requirement. Smooth, bit-stream-like traffic can be monitored much more effectively than highly variable and bursty traffic. Video services with variable bit rate show both short-term variations within a frame or between consecutive frames along with long-term variations, e.g., due to scene changes, with a time constant of several seconds. Both effects cannot be captured sufficiently by a simple policing mechanism. As shown above, it is possible to enforce a source close to its equivalent bandwidth rate with a relatively small time constant. However, from Figure 5.9, the ability of detecting violation is not satisfactory. Although it is expected that the output buffer can somehow shape the output bit rate; due to the stringent time constraint, efficient policing still needs some more efforts.

Based on above considerations, we propose a *Dual Leaky Bucket* mechanism as follows:

- i) First LB with moderate length counter enforces flow to its mean bandwidth, violating cell is marked as low priority cell.
- ii) Second LB with counter limit based on QOS requirement enforces flow within equivalent bandwidth, violating cell is discarded. The equivalent bandwidth is

calculated assuming single transmission, even statistical multiplexing is possible.

This design is supported by following arguments. With a moderate length counter, the first LB can effectively monitor the flow's mean rate over a relatively short time constant. Short-term fluctuations which can not be absorbed by the first LB are marked as low priority cells and sent into the network. However, if a good statistical multiplexing environment is preserved, most of these low priority cells will not be lost. This situation can be achieved by the second LB. The second LB with equivalent bandwidth policing monitors the traffic based on a counter limit which satisfies the QOS requirement. A well-behaved connection is guaranteed to have *cell discarding (loss) probability* within what is bargained. However, any connection which tries to overflow the link is punished by discarding every violating cell simply because it exceeds the allocated bandwidth. By rigorously regulating the flow at network access points, a well-multiplexed environment is maintained within the network and loss probability of low priority cells is expected to be small. Figure 5.10 shows that approximately 10% of the cells from a well-behaved connection (overload factor = 1) are marked as low priority based on mean bandwidth policing with  $N = 100$ . If the link is not overflowed by a misbehaved connection, whether or not these low priority cells will get lost greatly depends on the accuracy of bandwidth allocation. Figure 5.10 also indicates that when a connection intends to transmit at twice its declared mean rate, about 20% of the cells will be discarded at the access point; another 40% are marked as low priority cells. Only about 40% of the cells are regarded as high priority cells and receive full protection. When the overload factor increases to 3, almost half of the cells are discarded. This situation shall discourage anyone who

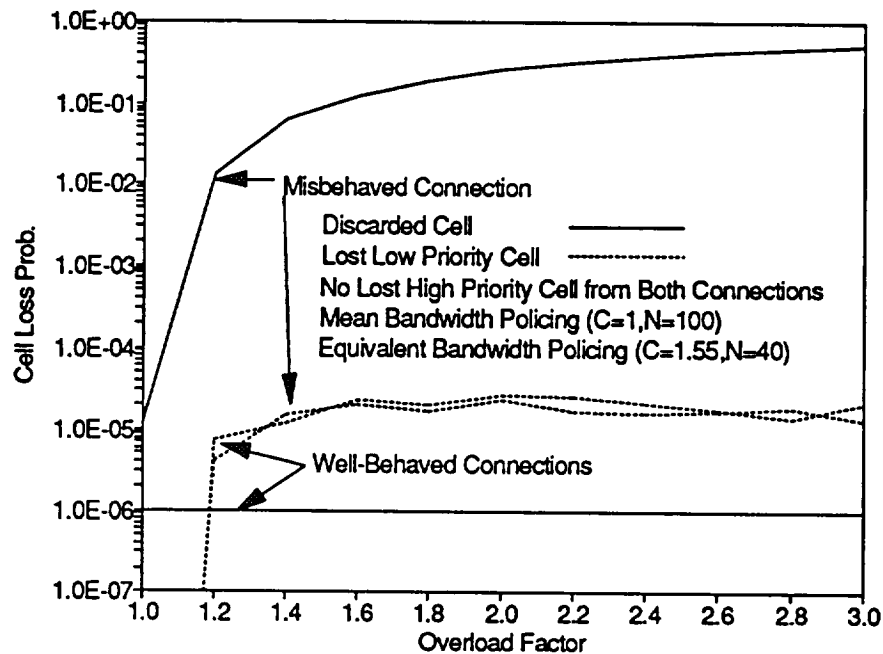


**Figure 5.10** Percentage of different cells for various overload factors using dual policing scheme.

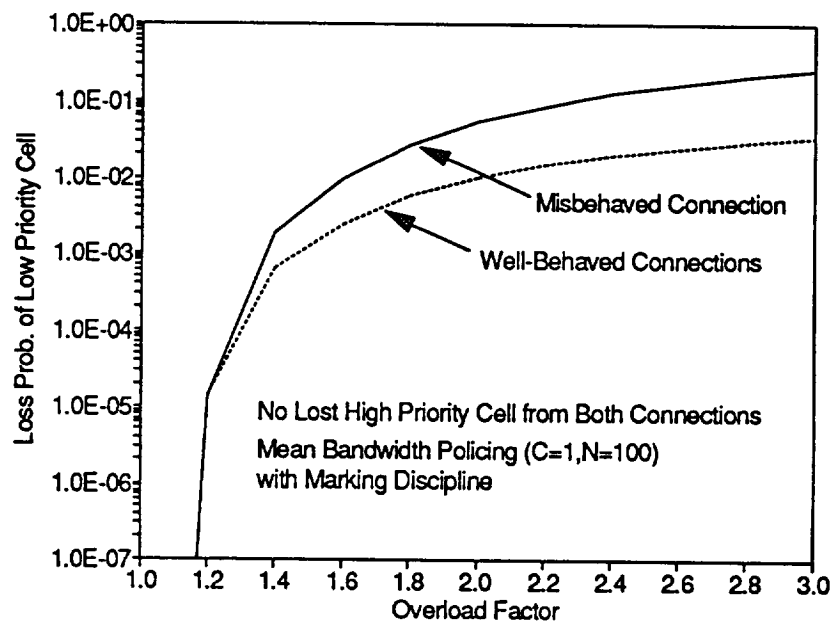
attempts to start a bad connection. Another possible and more drastic action is to disconnect the misbehaving transmission directly. It would be quite sufficient to judge a bad connection if the second LB counter exceeds counter limit over a pre-defined duration. In order to see that if a misbehaved connection is able to degrade the QOS of other concurrent well-behaved connections, simulations were performed using a *push-out* mechanism [63] with three homogeneous video transmissions multiplexed into a switching node. With such mechanism, an arriving high priority cell can enter a saturated queue provided that a low priority cell is already awaiting transmission. The low priority cell which spent least time in the queue is discarded and the high priority cell joins the queue in sequence. If the queue contains only high priority cells, the arriving high priority cell

is discarded. On the contrary, low priority cells cannot enter a full queue and are discarded. One of the connections was put in a misbehaving condition by multiplying the mean rate with an overload factor. The other two video transmissions were well-behaved connections. The link rate of the switching node is the equivalent bandwidth considering three multiplexed connections using Eq. (5.3). Buffer space is provided to meet the QOS requirement for approximately  $t_d = 10$  ms and  $p_l = 10^{-5}$ . Figure 5.11 displays that dual policing mechanism satisfactorily discards cells from the misbehaved connection up to about 50% when the overload factor is 3. The cell loss probability of a low priority cell from well-behaved connections is kept close to  $10^{-5}$ . No high priority cell from both connections is lost. Note that the total cells generating from mis/well-behaved connections are different considering the overload factor. Although Figure 5.11 shows the loss probabilities of a low priority cell from both connections are close, the amount of lost cells are quite different. The values for well-behaved connections are the average of two connections. No major difference is observed from these two well-behaved connections. Figure 5.12 shows the performance of a single LB mechanism with mean bandwidth policing and marking discipline, e.g. a *virtual* LB mechanism [59], under the same situation. It is noted that since the misbehaved connection pours a vast amount of low priority cells into the network (about 70%, from Figure 5.10, when the overload factor is 3), the loss probability of the low priority cell from the well-behaved connections exceeds  $10^{-2}$ , three orders of magnitude over the QOS requirement. Figure 5.13 demonstrates that a satisfactory result can also be achieved by using a single LB mechanism with equivalent bandwidth policing and discarding discipline. However, it can

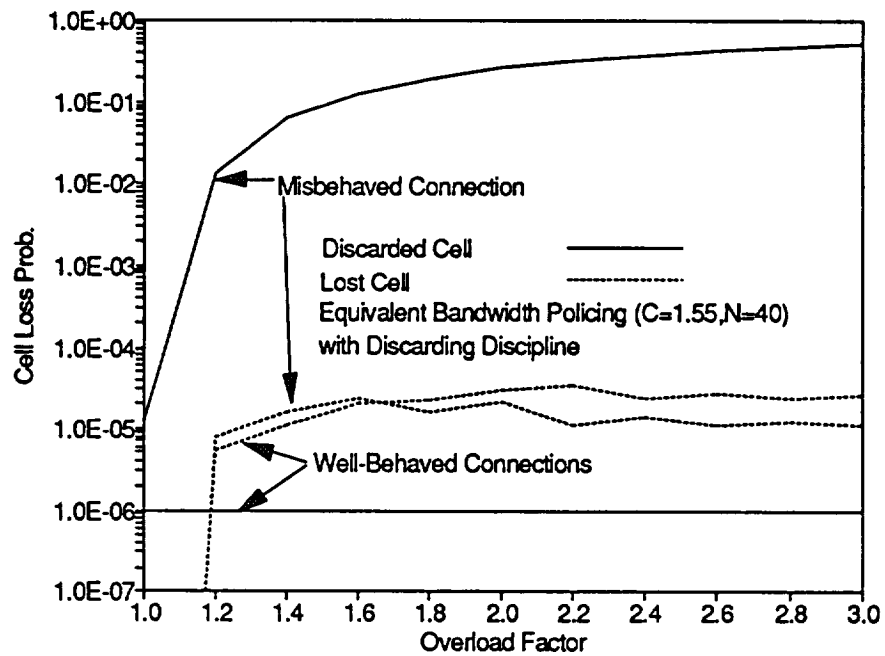




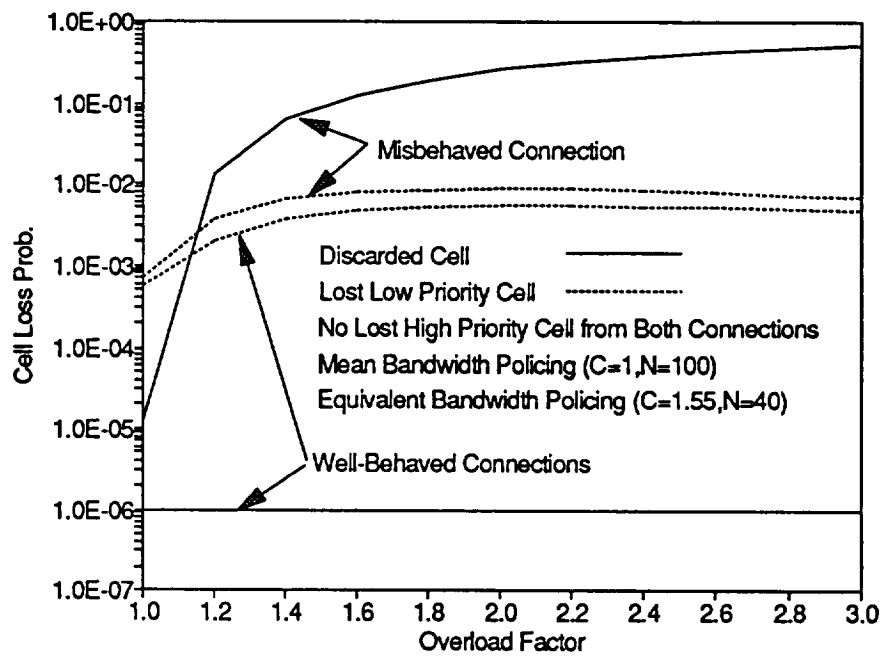
**Figure 5.11** Performance of dual policing mechanism under equivalent bandwidth allocation.



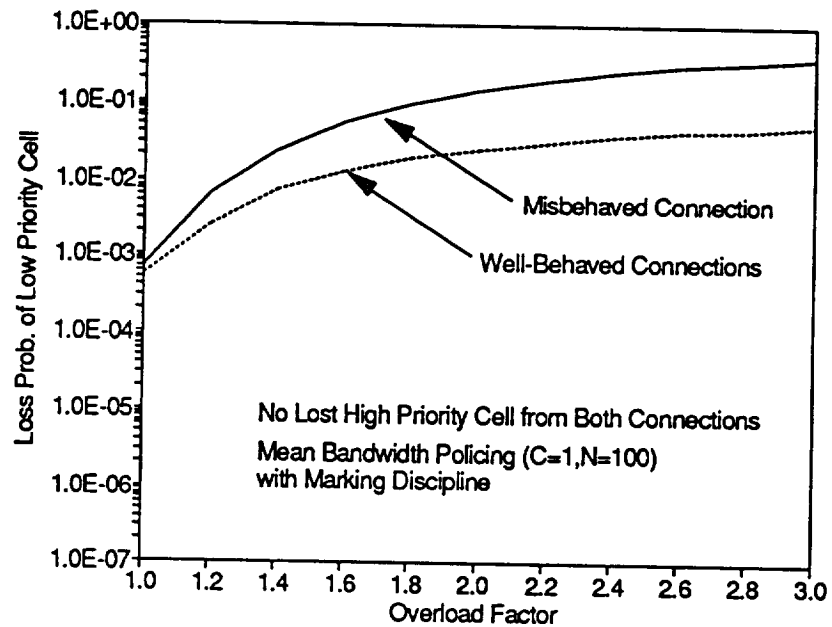
**Figure 5.12** Performance of mean bandwidth policing with marking discipline under equivalent bandwidth allocation.



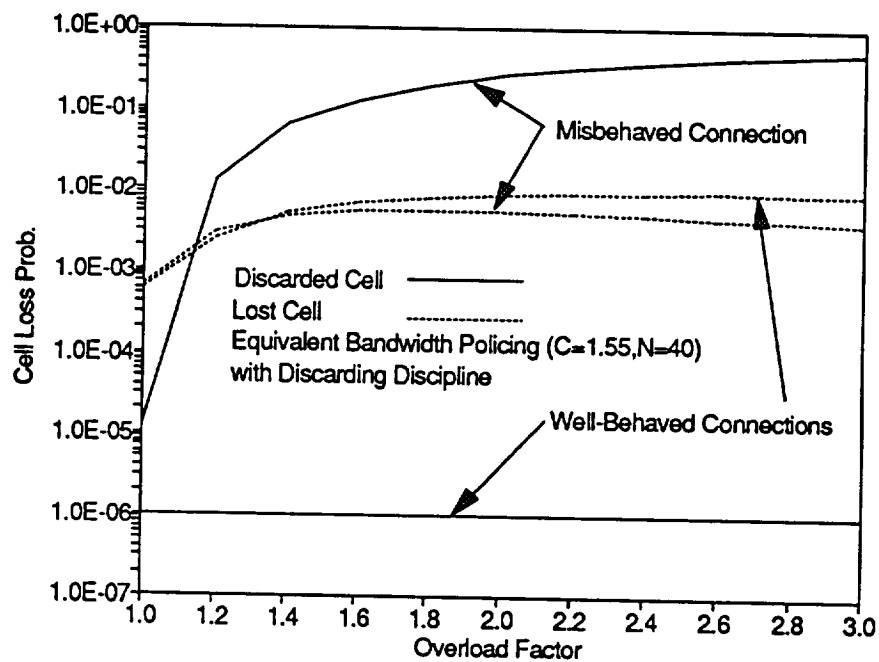
**Figure 5.13** Performance of equivalent bandwidth policing mechanism with discarding discipline under equivalent bandwidth allocation.



**Figure 5.14** Performance of dual policing mechanism under aggressive bandwidth allocation.



**Figure 5.15** Performance of mean bandwidth policing mechanism with marking discipline under aggressive bandwidth allocation.



**Figure 5.16** Performance of equivalent bandwidth policing mechanism with discarding discipline under aggressive bandwidth allocation.

be argued that the good performance probably comes from a conservative bandwidth allocation design. Remember that the equivalent bandwidth for our video source is overestimated since the burst/silence cell generating pattern is adopted. It is well known that cell-level congestion is determined by the call/burst level control approach. In order to test the sensitivity of the above three policing algorithms to the aggressiveness of the bandwidth allocation scheme, the link rate is decreased by 10%. The decrease of link rate represents a more aggressive bandwidth allocation or a higher utilization of the network. As shown in Figure 5.15, mean bandwidth policing with marking discipline shows slight sensitivity to bandwidth decrease. As expected, both connections experience some degree of increase in low priority cell loss. Comparing Figure 5.14 with Figure 5.16, it is clear that the dual LB policing scheme has a better performance in protecting good connections. There is almost one order of magnitude difference in cell loss probability between the two policing algorithms when the overload factor is relatively large.

The LB mechanism is easy to implement by using only two counters, one for counter state and one for the measurement of the decrementation interval. Therefore, the dual LB policing algorithm does not need too much additional effort. As shown above, it is an efficient policing scheme in both normal and overload situations. Particularly with somewhat inaccurate or aggressive call admission control, it has much better performance than other policing schemes. It is noted that the superior performance of the dual LB mechanism is not simply because of the increased bucket number but also because of the nice combination of the call admission control and the policing function. Meanwhile, different actions enforced on violating cells from two buckets can effectively absorb

short-term fluctuations and detect long-term overloading at the same time.

### 5.3 Priority Control and Selective Discard Mechanism

Due to the diverse characteristics of traffic streams competing for network resources, some form of prioritization must be in place to determine how cells should be treated in the network. Obviously, cells from connections that require a continuous per-connection QOS, in terms of cell delay and cell loss, must be given preference to those that can tolerate a long-term average QOS. Keep in mind that network provisioning and connection admission procedures will determine the degree to which cell-level congestion is likely to occur [42]. For a conservative connection admission policy or sufficient network resource situation, the chance of cell-level congestion is rare and the need for cell prioritization is negligible. Alternatively, if an aggressive admission policy is adopted, cell prioritization is then critical in ensuring that different connections receive their negotiated QOS. Some papers devoted to priority schemes and optimal discarding principles in ATM networks are [61,62,63,64].

The cells from connections which have reserved bandwidth allocation must receive the highest priority [39]. It is assumed that these connections have the most stringent QOS requirements, and hence their cells should not be affected in any way whatsoever from ATM cell-level congestion. The cells with the next highest priority are those designated as lower priority from connections with a per-connection QOS. This designation may result from the application marking the cell's CLP bit or the network tagging the cell because its arrival violated the connection's contract. Note that these cells must be placed

in the same buffers as their corresponding high-priority cells to maintain cell sequence order, however, these cells must be discarded early enough to ensure that sufficient space is available for arriving high-priority cells. The final cell-level priority is for connections that have an aggregate QOS. The unique cell loss rate requirement is now split into two parts: a more restrictive constraint on the loss of precious cells and a less restrictive constraint on the loss of ordinary cells. This split of cell loss rate requirement can improve transmission efficiency and network utilization. However, this is paid for by priority marking in the terminals and more complex buffer management logic in the switches and multiplexers.

A layered coding scheme has the ability to distinguish a given cell in a same connection to be a high or low priority cell. A high priority cell carries the most important information of the signal and must reach its destination. On the contrary, the loss of low priority cells will not harm the quality much. If the dual LB policing mechanism is adopted, by voluntarily marking some of the cells as low priority by coder, these cells are then automatically excluded from the calculation of the number of arriving cells in the first LB counter. But these low priority cells still have to be considered in the second LB counter in order to comply with the allocated bandwidth. If violation occurs in the second LB, low priority cells should be discarded instead of high priority cells using a selective discard mechanism. It is noted that if violations occur in the second LB counter, it is almost sure that sooner or later there will be violations in the first LB counter. Therefore, some of the original high priority cells will be marked as low priority. In order to avoid high priority cells which carry vital information getting discarded, the

coder has to decide the high/low priority ratio with caution. Basically, the transmission rate of high priority cells should not exceed the mean cell rate. Meanwhile, setting a less important cell as a low priority cell allows the cells to immediately enter the network without any further delay, at the cost of a potentially higher loss probability in the network. In next chapter, the CLP bit will be set up in the coding scheme to improve coding efficiency. Also some error control procedures will be examined to combat the higher loss probability of low priority cells.

## 5.4 Explicit Congestion Notification (ECN)

Network congestion is measured at the intermediate nodes along the path of the connection. The onset of congestion can be identified by either the cell arrival rate or the number of cells in the switching queue. ECN is a mechanism by which the end system is kept informed about the congestion status of the network. This is achieved by each network element continuously sending its congestion information to the end points of the connections that have traffic passing through it. To avoid generating extra cells during congestion, it has been proposed that this information be carried as a single bit indicator in the cell header [38]. Two principles govern the behavior of the ECN: *congestion detection* and *end system reaction*. Each node detects congestion by monitoring its buffer occupancies and trunk utilization. Upon determining that congestion may occur in the near future, the node sets an indicator in the ECN field of all passing cells whose connections are likely to be affected. Once the risk of congestion is released, this congestion indicator is reset appropriately. In response to the network congestion indicator, all sources switch

to low rate, resulting in a decrease in the queue length after a roundtrip feedback delay. The duration of the congestion depends on the traffic condition, link capacity and statistical gain.

Note that ECN is a reactive control scheme and could be very effective in LANs where propagation delay is small. As for high-speed WANs, ECN probably has to be implemented along with other preventive control schemes and serve as a backup system. Current research has shown that ECN is highly beneficial when the duration of congestion is at least an order of magnitude larger than the propagation delay [65]. ECN can be used in a variable bit rate codec for changing the coding mode to produce a lower bit rate output when ECN is indicated. Similar to ECN, the receiver can send back the information of cell delay which reflects the status of the intermediate node on the path. Based on this information, some suitable approaches which prevent cell loss will be addressed in next chapter.

## 5.5 Traffic Shaping

Traffic shaping, also known as *source rate control* or *smoothing function*, is a mechanism which changes the traffic's characteristics at user's own interests. As discussed in Section 5.2, most of the proposed policing functions cannot effectively control other source parameters than the peak rate [36]. Unlike policing functions which work on the network side, shaping functions work on the user side. Therefore, it is possible to have more controls on the characteristics than any policing functions. By using a shaping function, a codec is able to control the traffic parameters, typically the cell's minimum inter-



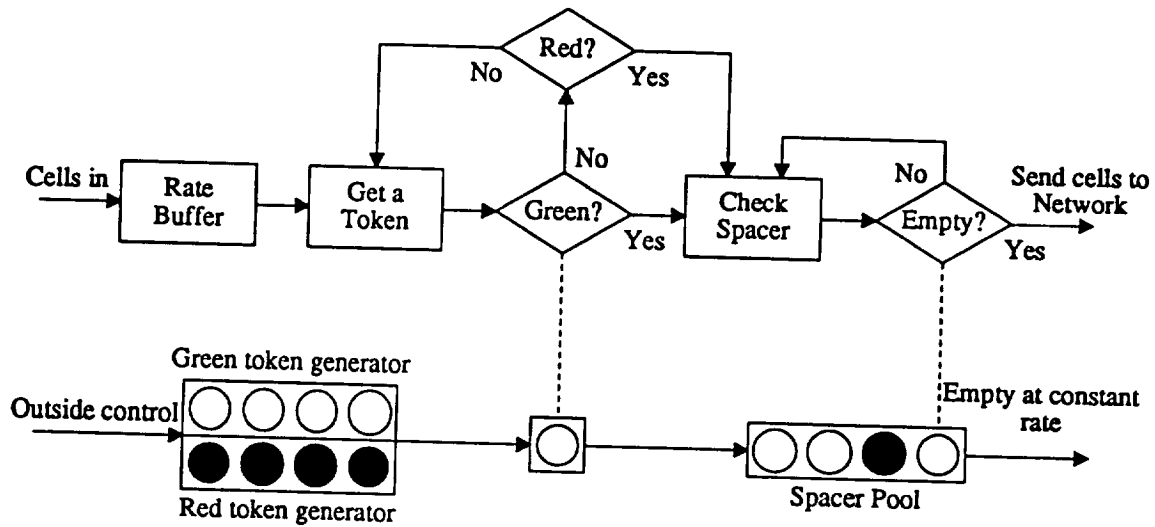


Figure 5.17 A prioritized traffic shaping function.

departure time (maximum bit rates) and the maximum source activity (fraction of time during which the source transmits) allowed in a given period [66].

Figure 5.17 shows a variant of the LB mechanism which can be implemented as a traffic shaper. This mechanism has some parameters to be controlled. The first parameter is the size of the bucket, which determines the maximum number of cells that can be sent for a period of time (burst length). An arriving cell that finds the token pool non-empty, departs immediately (in case that spacer pool is empty if a spacer is included) and one token is removed from the token pool and enters the spacer pool. Another parameter is the token generating rate which determines the long-term bandwidth available to the connection. The rate buffer is used to queue cells, which cannot immediately enter the network upon their arrival because of empty token pool. Two different token generators generate green and red tokens for high and low priority cells respectively, if a priority

scheme is desired. A spacer is included to add spacing between two cells that come too close to each other. The peak rate limitation can be achieved by requiring a minimum spacing between consecutive cells. Cell can only enter the network if the spacer pool is empty. The spacer pool is emptied at a constant rate which determines the minimum space between cells. The problem of setting these parameters depends on the QOS requirement and the allocated bandwidth. For example, the probability of losing cells at the access point is a function of the combined token pool and rate buffer size [56]; smoothness (increases if token pool size is decreased) of traffic is a trade-off with delay.

## 5.6 Some Notes

Note that the equivalent bandwidth calculated for video transmission in this chapter is overestimated since a burst/silent cell generating pattern within a frame is assumed. From the simulation results, the proposed dual leaky bucket algorithm has better performance than other policing schemes in protecting good connections and punishing misbehaved connections. Furthermore, it is also easy to implement. It is noted that the equivalent bandwidth used in dual leaky bucket mechanism can be replaced with “allocated bandwidth” if another call admission scheme is adopted in future ATM networks; the same arguments and results still hold. Other congestion control approaches which have influence in video codec design have also been investigated. The primary objective of this chapter intends to close the gap which lays between network protocol and video codec design. Each congestion control mechanism described above interacts with some elements of video codec. Based on the understanding of these interactions, an efficient prioritized

video transmission scheme which follows the concept of dual leaky bucket mechanism will be proposed and studied in next chapter.

# Chapter 6

## Video Codec Design

Our ultimate goal is to design a video compression scheme which can obtain not only efficient cost/quality application but can also adjust appropriately to the network status. Taking the network's characteristics into account, a complete set of design principles for video codec in packet video environment is proposed in this chapter. The principles follow the concept of the proposed dual leaky bucket mechanism, and a prioritized video transmission scheme is developed which has general application in any video compression encoder including the four specific coding approaches introduced in Chapter 3. The coding behavior is investigated under the environment which we assume to be similar to that in future ATM-based B-ISDN networks. All the characteristics and mechanisms introduced in Chapter 2 and 5 are included for consideration. Finally, despite the unavoidable nature of cell loss in ATM networks, we would like to evaluate an error recovery scheme which can improve the overall video quality.

### 6.1 Call Setup

The call setup procedure is composed of two phases [56]. In the first phase, the source of the call notifies the destination and the intermediate nodes along the path of the new call and its characteristics. This phase is accomplished by the source sending a request message to the destination. The second phase includes a call confirmation process in which a confirmation message is transferred from the intermediate nodes back to the source if the requested capacity is available. If any node along the path does not have enough capacity, it sends an abort message back to the source. Keep in mind that the nodes along the routing path may have different traffic conditions, therefore the bandwidth needed to support the QOS of the requesting call in every node is possibly different. For example, if there are 10 multiplexable video transmissions which go through node *A* and only 5 go through node *B*, then for a video transmission which is routed to travel through node *A* and *B*, the equivalent bandwidth needed in node *A* is quite possibly less than the equivalent bandwidth needed in node *B* due to the gain of multiplexing. The available capacity of specific nodes is contained both in confirmation and abort messages. On receipt of one abort message, the call is said to be blocked and the source can try later in order to maintain desired QOS. Or it can resend a setup message according to the minimal bandwidth it got from all intermediate nodes by sacrificing transmission quality. Following the call admission procedure described in the previous chapter, we need to define a traffic metric  $(R_{peak}, \rho, b)$  which is carried in the request message to describe the call. The peak transmission rate of a video codec is determined by hardware design and is a constant. However, the peak rate of a particular transmission depends on the contents of video and quality desired, and is usually below the codec's peak transmission speed

	$R_{peak}$	$\rho$	$b$	$R_{mean}$	$\sigma$	EB1	EB5	EB10	EB20
Seq 1	1.24	0.763	0.025	0.95	0.029	1.07 (13%)	5.05 (6%)	9.93 (4%)	19.61 (3%)
Seq 2	2.29	0.833	0.027	1.91	0.050	2.11 (10%)	10.07 (5%)	19.84 (4%)	39.26 (3%)
Seq 3	3.36	0.308	0.010	1.04	0.612	2.62 (152%)	11.49 (121%)	19.30 (85%)	33.39 (60%)
Seq 4	4.28	0.892	0.029	3.83	0.171	4.10 (7%)	20.52 (7%)	40.78 (6%)	80.12 (5%)
Seq 5	1.55	0.763	0.025	1.19	0.110	1.36 (15%)	6.84 (15%)	13.50 (13%)	26.06 (9%)
Seq 6	5.22	0.847	0.028	4.43	0.450	4.99 (13%)	24.99 (13%)	49.99 (13%)	97.86 (10%)
Seq 7	7.67	0.361	0.012	2.77	0.620	6.99 (152%)	20.23 (46%)	36.72 (32%)	68.16 (23%)
Seq 7a	3.76	0.736	0.024	2.77	0.620	3.48 (25%)	17.40 (25%)	34.80 (25%)	68.16 (23%)
Seq 8	12.72	0.465	0.015	5.92	1.000	12.18 (105%)	39.89 (34%)	73.75 (25%)	138.98 (17%)
Seq 8a	7.61	0.778	0.026	5.92	1.000	7.32 (24%)	36.60 (24%)	73.20 (24%)	138.98 (17%)

unit: Mbits/sec

 $\rho$ : utilization, fraction of time source is active and transmits at  $R_{peak}$  $b$ : average duration of an active period (second) $\sigma$ : standard deviation of bit rate

EBn: equivalent bandwidth with n multiplexed homogeneous video sources

Seq 1: "Susie", H.261, MC\_on, p=15, T=1

Seq 2: "Football", H.261, MC\_on, p=30, T=1

Seq 3: "Susie", ADTV, C=0.96, T=1, QS=4

Seq 4: "Football", ADTV, C=3.84, T=1

Seq 5: "Susie", subband

Seq 6: "Football", subband

Seq 7(a): "Susie", MC\_on, T1=10, T2=5 (alternative declaration)

Seq 8(a): "Football", MC\_on, T1=25, T2=25 (alternative declaration)

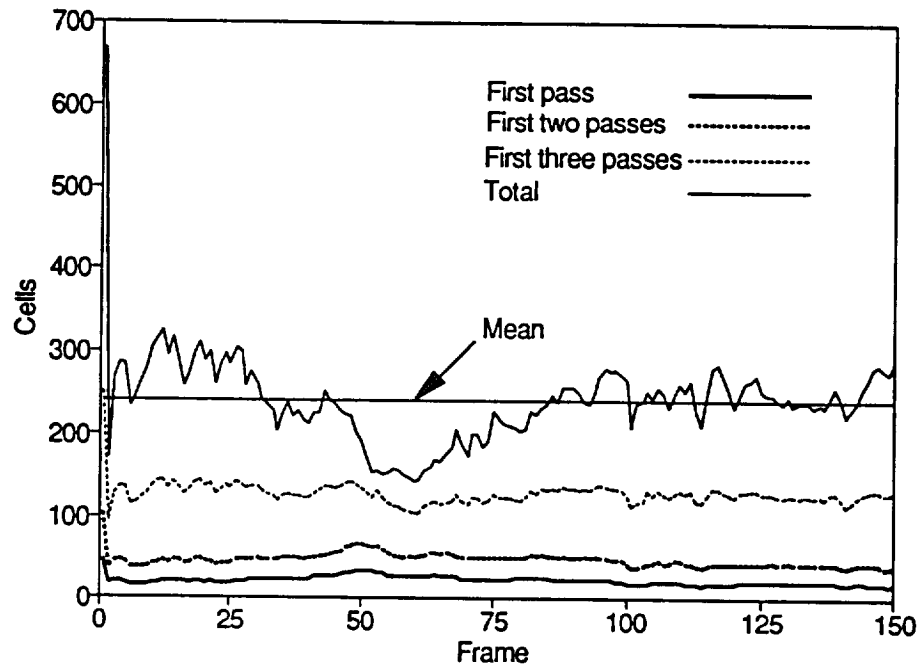
%: percentage of equivalent bandwidth over mean bit rate

Table 6.1 Traffic metric and equivalent bandwidth for several video sequences.

and can be controlled by traffic shaper. Table 6.1 shows the traffic metric and equivalent bandwidth for several coded sequences from Chapter 3. The equivalent bandwidth is calculated with Eq. (5.3). Cell loss probability  $p_l$  is  $10^{-5}$  and the buffer space at intermediate nodes along the routing path is set to be 256 cells. Sequences 1 and 2 are coded with the H.261 algorithm and their bit rates are quite constant because of the regulation of the rate buffer. Thus, the equivalent bandwidth is very close to the mean bit rate. In fact, they are CBR transmissions since the H.261 algorithm is designed for a specific bandwidth channel (px64 kbits). Sequences 3 and 4 are generated using the ADTV technique. For Sequence 3, the equivalent bandwidth without multiplexing is 152% over the mean bit rate. If there are 20 multiplexed homogeneous connections, the equivalent bandwidth is down to 60% over mean bit rate, which is still considered high. Although the coding output rate is quite bursty, there are two mitigating circumstances:

1. the pattern of burstiness is relatively "uniform". That is, the data rate peaks every 13<sup>th</sup> frame by design.
2. the variations occur very fast, that is high traffic persists for only a single frame followed by low traffic.

Because of (2) the traffic can be smoothed out using a moderate sized buffer, and (1) implies that the size of the rate buffer can be ascertained with some confidence. Sequences 5 and 6 are processed with subband coding and the bit rates are relatively smooth. Sequences 7 and 8 are coded with the MBCPT coding scheme. It is observed that the equivalent bandwidth is close to the peak rate and much greater than the mean rate.



**Figure 6.1** Cell distribution of 4 passes for Sequence 7.

This phenomenon is attributable to the high peak rate of one single frame. Although the billing procedure in ATM networks is not determined right now, the peak rate is likely to play an important role. With such a bandwidth allocation, it is clear that a lot of reserved bandwidth will be wasted and result in inefficient transmission. Figures 3.19 and 3.25 shows that the peak rate occurs at the very first frame which is intra-mode coded. Intra-mode coding takes away the advantage of temporal prediction and thus generates a much higher data rate. From Table 3.6 the PARs are 2.77 and 2.15 for Sequences 7 and 8 respectively. However, intra-mode coding is inevitable because of its importance in synchronization and prevention of error propagation. Several approaches can be considered to handle the intra-mode coding frame:



- **Coding with a lower quality:** This approach is questionable since the intra-mode coding frame usually serves as the anchor frame for motion compensation. Degrading the intra-mode coding frame will hurt the quality of the entire sequence.
- **Smoothing with a rate buffer:** Buffer space is limited in order to avoid large delay.
- **Marking excess cells as low priority:** As shown in Figure 6.1, data from the fourth pass constitutes the excess cells over the mean rate. However, a large amount of cell loss is undesirable since again the quality of anchor frame is critical.

The three approaches described above can be used together to achieve the desired results. This is shown in simulation results of Section 6.7. Taking Sequence 7 as an example, not considering the first frame, the next highest rate occurs at the 12<sup>th</sup> frame with 1.48 bits/pixel. Now we claim the traffic metric as (3.76 Mbits/s, 0.736, 0.024 sec) and achieve an equivalent bandwidth which equals to 3.48 Mbits/s; a reduction of over 50% from 6.99 Mbits/s. It is clear that the peak rate description is critical in determining equivalent bandwidth. Meanwhile, because MBCPT is variable bit rate coding scheme, it not only obtains a smooth quality sequence, but also a very multiplexable output pattern. From Table 6.1 it is observed that when there are 20 homogeneous video sources multiplexed together, the percentage of equivalent bandwidth over mean rate goes down to 23 and 17 for the *Susie* and *Football* sequences, respectively. Figure 6.2 shows a general video codec. The detailed function for each element is given in following sections.

## 6.2 Design A Traffic Shaper

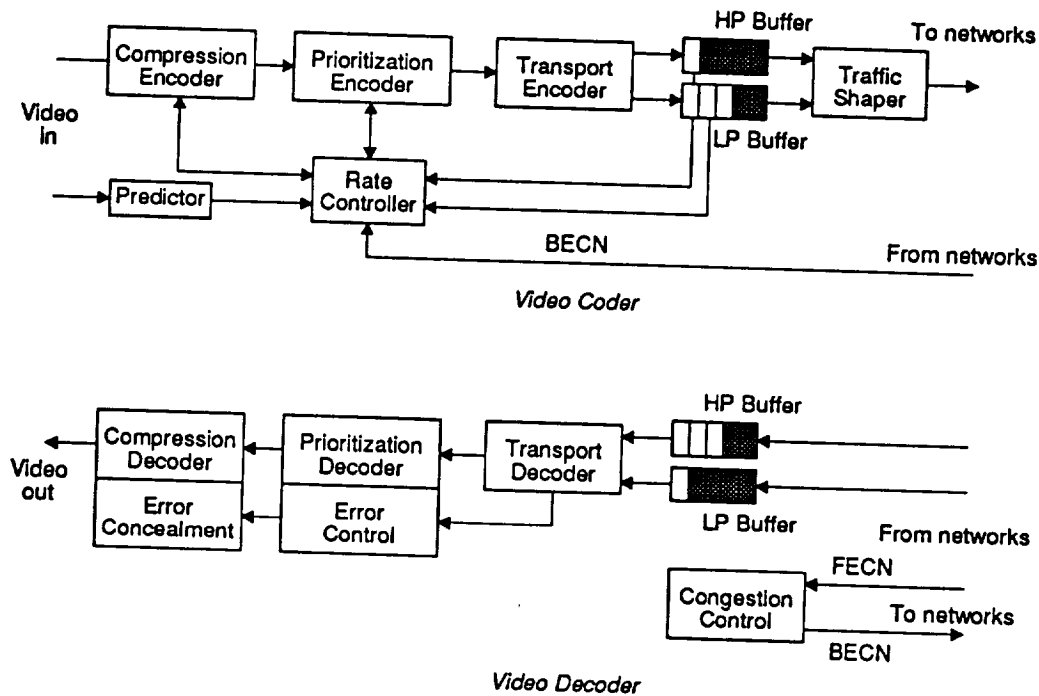


Figure 6.2 A general video codec.

A leaky bucket is implemented in the codec as a traffic shaper and works as the counterpart of the policing function in network [43]. By implementing a traffic shaper, we can assume thereafter that the transmission is a well-behaved connection and no cell from this connection will be deemed in violation by policing function (the delay jitter between shaper and network policing location is neglected). We would like to relate the leaky bucket parameters to the bandwidth allocation procedure described in the previous section. First, the token generating rate  $\gamma$  determines the long-term average bandwidth available to the connection and in general it should never exceed the equivalent bandwidth. Violating this condition could result in network congestion. In our simulation, *green* and *red* tokens are generated at rate  $\gamma_{grn}$  and  $\gamma_{red}$  for HP/LP cells respectively. Several options

for determining  $\gamma_{grn}$  and  $\gamma_{red}$  can be considered for best results:

- $\gamma_{grn}$  = equivalent bandwidth,  $\gamma_{red}$  = certain fraction of equivalent bandwidth [56]

This approach takes full advantage of equivalent bandwidth. In the absence of a green token, a red (LP) cell can be sent into the network to avoid further delay. Adopting this option, the traffic shaper must be well designed to avoid green cells being marked “red” by UPC and experiencing high cell loss probability. Also the network does not guarantee the QOS of red cells in this case.

- $\gamma_{grn}$  = mean bandwidth,  $\gamma_{red}$  = equivalent bandwidth - mean bandwidth

This option follows the concept of the dual LB mechanism and guarantees the priority assignment will not be altered by the UPC. It also guarantees the QOS of red cells in a statistical multiplexing environment.

- $\gamma_{grn}$  and  $\gamma_{red}$  are adjusted dynamically

Token generating rates  $\gamma_{grn}$  and  $\gamma_{red}$  may vary with the frame type. For instance, the I, P, and B frames in ADTV coding scheme can be assigned with different  $\gamma_{grn}$  /  $\gamma_{red}$  ratio under the constraint  $\gamma_{grn} + \gamma_{red} \leq$  equivalent bandwidth.

The second option is adopted in our simulator. Once  $\gamma$  has been set, the token pool size  $M$  can be computed to ensure a given desired access delay probability. Based on fluid-flow representation of the leaky bucket system, the token pool size  $M$  is given by [68]:

$$M = \frac{b(1 - \rho)\gamma(R_{peak} - \gamma)}{\gamma - \rho R_{peak}} \ln \left[ \frac{(\gamma - \rho R_{peak}) + \rho \xi (R_{peak} - \gamma)}{\xi \gamma (1 - \rho)} \right] \quad (6.1)$$

where  $(R_{peak}, \rho, b)$  is the traffic metric,  $\gamma$  is the token generating rate ( $= \gamma_{grn} + \gamma_{red}$ ) of the

connection, and  $\xi$  is the desired access delay probability which determines the probability that an arriving cell will find an empty token pool.

The peak rate of cells entering the network is controlled through the use of a spacing function. An arriving cell which has already obtained a token can only enter the network if the spacer pool is empty. The spacer pool is emptied at a constant rate  $\beta$  where  $\gamma \leq \beta \leq R_{peak}$ . In our simulator,  $\beta$  is equal to  $\gamma$  for maximum smoothing (at the cost of increased access delay).

### 6.3 Design A Rate Buffer

The rate buffer has two main functions. An arriving cell which finds an empty token pool is placed in the rate buffer. After the token pool size  $M$  is determined, the rate buffer size  $B$  can be dimensioned to ensure an access loss probability for a well-behaved source; a source which generates output flow according to its declared traffic metric. Another function of the rate buffer is sending the buffer status back to the rate controller. The rate controller then can decide upon a coding strategy depending on buffer fullness. In our simulator, two rate buffers, namely the HP and LP buffers, are implemented to hold high

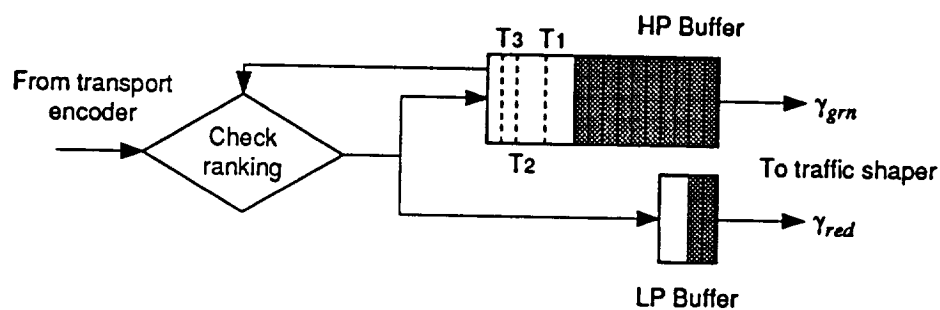
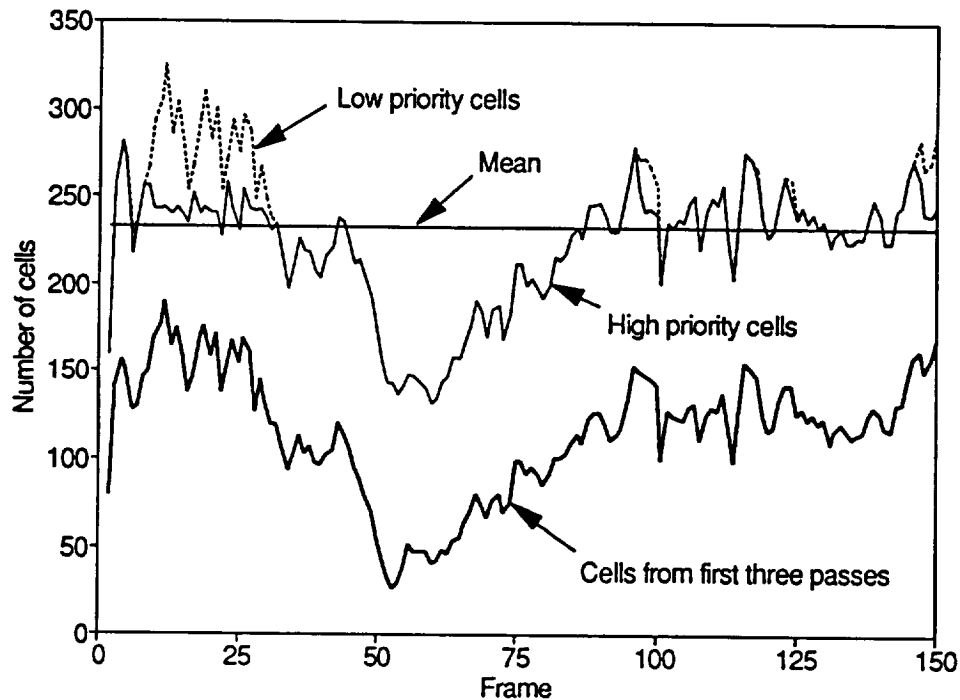


Figure 6.3 A rate buffer with priority mechanism.



**Figure 6.4** Distribution of high/low priority cells for Sequence 7a.

and low priority cells. Three thresholds,  $T_1$ ,  $T_2$ , and  $T_3$ , are set up in the HP buffer and determine which buffer a cell should enter. As will be defined in Section 6.5, a cell is assigned one of four priority ranks depending on its importance. If an arriving cell with rank 4 finds HP buffer fullness exceeding  $T_1$ , it has to enter the LP buffer and is regarded as a low priority cell. A rank 3 cell has to enter LP buffer if HP buffer fullness is over  $T_2$ . Only a top ranked cell can enter the HP buffer when the buffer fullness exceeds  $T_3$ . Figure 6.3 demonstrates such a mechanism. Because two buffers have different output rates, it is reasonable to have HP/LP buffer sizes in proportion to their output rates. Then cells entering different buffers will experience the same maximum access delay which is important to decrease delay jitter. Considering short-term fluctuations of the cell arrival process for the four priority rank cells, thresholds are chosen to take full advantage of the

HP buffer without blocking high rank cells from entering the HP buffer. Figure 6.4 shows the distribution of HP/LP cells for Sequence 7a. It is clear that, with such a design, low priority cells all come from pass 4 data which is insignificant. In this simulation,  $B_{HP}$  and  $B_{LP}$  are set to be 128 and 33 in order to preclude any access cell loss. Thresholds  $T_1$ ,  $T_2$ , and  $T_3$  are 96, 112, 120 respectively.

## 6.4 Packetization

The function implemented in transport processor segments video information, coding mode information, if it exists, and synchronization information, into transmission cells. In order to prevent the propagation of an error resulting from cell loss, it is desirable to make cells independent of each other. This means no data from the same block or same frame is separated into different cells. However, it is possible to include chaining and

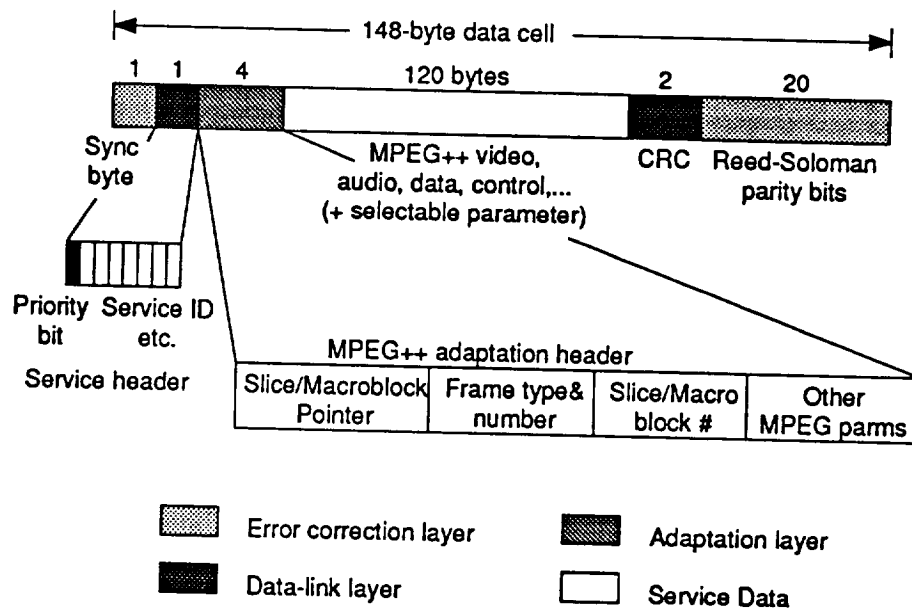


Figure 6.5 Data format of ADTV transport cell.

segmentation information in the coding header; then “*data groups*” are allowed to be segmented across cells. This feature provides a high degree of efficiency in the use of fixed-size cells while limiting the propagation of channel error from one cell to the next. In order to limit the delay of packetization, it is necessary to stuff the cell with dummy bits if the cell is not completely full within some time.

A complete transport layer of the ADTV system has been developed by *Advanced Television Research Consortium* (ATRC) [73]. Since the transport layer plays a key role in video transmission, we introduce that layer of the ADTV system in the following as an illustration. The transport layer of ADTV is a packet-oriented approach to reliable video delivery. It consists of three distinct sublayers: “data link level”, “adaptation level” and “video service level”. The data format is shown in Figure 6.5. After formatting, both high and standard priority bitstreams (two priority classes, namely high and standard, are defined in the ADTV system) in ADTV appear as sequences of fixed length 148-byte cells (remember that the length of an ATM cell is 53 bytes), each containing 120 bytes of service data (“payload”), a 1-byte synchronization header, a 1-byte data-link sublayer header, a 4-byte adaptation level header, a 2-byte cyclic redundancy check (CRC) trailer and a 20-byte Reed Solomon parity trailer. It is claimed that the fixed-length cell structure is able to achieve good error protection and rugged data synchronization under fluctuating channel conditions, while providing service-specific adaptation facilities for logical resynchronization after uncorrectable error events.

#### *Transport Encoder*

The data link sublayer is based on a “cell relay” asynchronous time multiplexing concept

which is similar to ATM. The data-link header contains information such as a priority indicator, a service ID and a cell sequence number and is intended to provide service-independent transport services such as priority support, service multiplexing, and cell-error detection and correction. The video-specific adaptation sublayer has the function of efficiently packing variable length ADTV data into fixed length cells, while supporting rapid decoder recovery in the event that one or more cells are received in error. Thus, the adaptation headers contain information such as frame type indicators, slice (a set of integer number of adjacent macroblocks) /macroblock IDs, priority breakpoint and re-entry pointers (used to classify HP/SP data) needed to support segmentation, chaining and error control at the video decoder.

The data-link sublayer is implemented in the form of a time-division multiplexer which adds suitable headers to video, audio, data and control, and then services them in a predetermined manner. These high and standard-priority bitstreams are finally processed by an error control module for addition of CRC and Reed-Solomon parity bits before entering the channel.

#### *Transport Decoder*

For each of the high and standard priority bitstreams received from the channel demodulator, the transport decoder performs Reed-Solomon decoding and CRC based error detection. Error corrected and detected cells are forwarded to a data link level demultiplexer for splitting into individual service streams including video, audio, data and control. Cells received in error are not processed by the demultiplexer. The adaptation sublayer decoder is responsible for logical resynchronization of ADTV video decoding.



When error events occur, the transport decoder passes on an appropriate error condition and resynchronization information to the priority decoder.

## 6.5 Design A Priority Scheme

The main function of the prioritization process is to assign priority to data element. The approach of priority assignment is equivalent to an asynchronous *codeword* multiplex scheme in which each codeword or data element is multiplexed to one of the priority classes according to the assigned priority for that data element. First these data elements are ranked in terms of their relative importance and then priority is assigned depending on the HP/LP buffer fullness. There is a natural way to assign priority rank to coding data of layered coding schemes, like MBCPT and subband coding, since they discern the importance of data elements through the coding procedure. For MBCPT, four priority ranks are assigned as

1. *Headers, motion vectors, and first pass coding data*
2. *Second pass coding data*
3. *Third pass coding data*
4. *Fourth pass coding data*

As for H.261 and ADTV, which are based on DCT, the natural priority rank is

1. *Headers*
2. *MB address, types and quant*

3. *Motion vectors*
4. *DC values*
5. *Low frequency coefficients*
6. *High frequency coefficients*

The priority decoder has to perform the function of reassembling a single video bitstream from the received HP/LP cells.

## 6.6 Adaptive Coding Based on Network Status

Based on an ECN/CLP combination, networks (or a specific transmission link) can be defined as following four states [38]:

1. *Normal state*:  $ECN = 0$  and no cell loss.
2. *Slight congested state*:  $ECN = 1$  in cell( $CLP = 1$ ) and no cell loss.
3. *Medium congested state*:  $ECN = 1$  in cell( $CLP = 1$ ) and loss of cell( $CLP = 1$ ).
4. *Heavy congested state*:  $ECN = 1$  in cell( $CLP = 0$ ).

It is reasonable to assume that a high priority cell ( $CLP = 0$ ) will not be discarded by appropriate call admission control and usage parameter control. The above mechanism is accomplished by each intermediate node appropriately monitoring its queue occupancy. Considering a set of three thresholds,  $T_1 < T_2 < T_3$ : a cell( $CLP = 1$ ) has its ECN set to 1 if congestion exceeds  $T_1$ ; a cell( $CLP = 1$ ) is discarded if congestion exceeds  $T_2$ ; and a cell( $CLP = 0$ ) has its ECN set to 1 if congestion exceeds  $T_3$ . The destination node can

conclude the network status by above information and send a BECN (backward ECN) back to the source node.

According to different network states, rate buffer fullness, and either a pre-calculated information index (single frame) or AR model prediction (multiple frames) [26,31], the video coder has to decide a strategy for best coding/transmission results. Facing a possible network congestion, the video source may need to reduce its output rate to help the network to release the congested situation (it is quite possible that the network will force every source to do so by changing UPC parameters) [40]. The video coder can dynamically adjust its output flow by choosing the coding thresholds, motion compensation thresholds, quantizer stepsize or simply increasing interleaving. It may also change the HP/LP cell ratio and let the network discard cells selectively when it is necessary.

## 6.7 Error Control

Actually there is no way to guarantee that cells will not get lost after being sent into the network. In a packet video environment, we are generally concerned with *cell loss* rather than *symbol error*. Cell loss can be mainly attributed to three reasons:

1. Bit errors occur in the VPI/VCI field, leading the cells astray in the network.

However, in fiber-based networks, bit error rate is quite small. Also the ATM header contains Header Error Check (HEC), which is a CRC field, to provide error protection for minimum misrouting. Therefore, cell loss due to this reason is

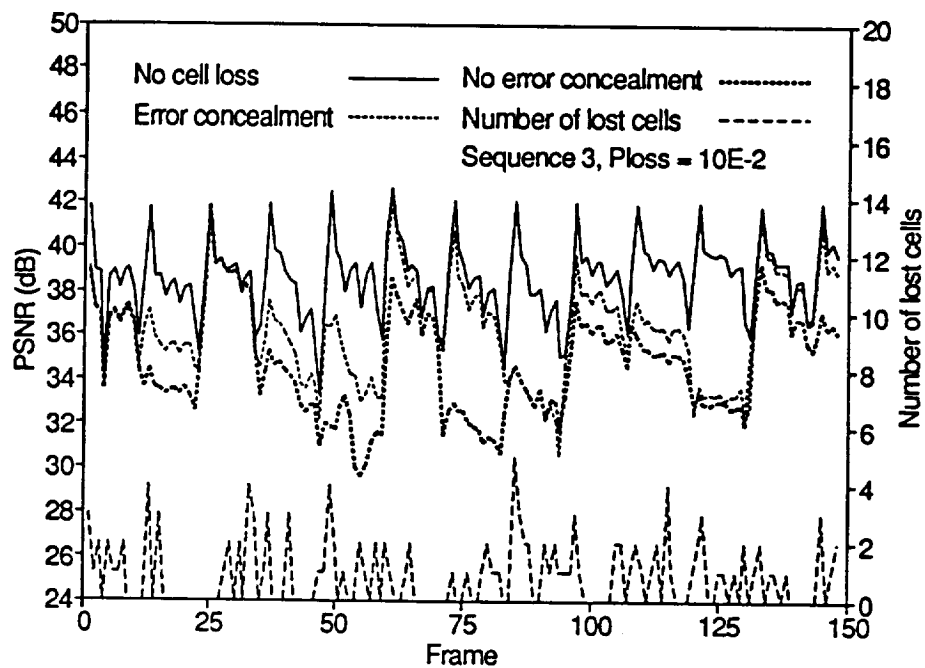
negligible.

2. The network becomes congested resulting in a reduction of available bandwidth, which results in an increase in cell delay and cell loss.
3. The output of video coder increases well above the negotiated capacity which results in access loss.

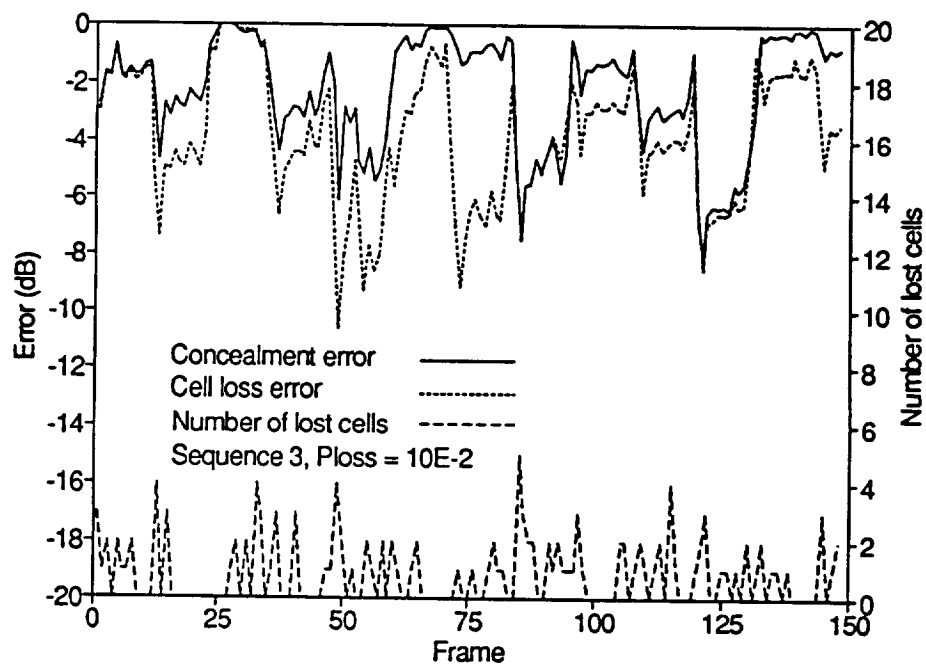
Effects created by ordinary data loss are masked by the vital data and have minor effect when viewing at video rates because the lost area is scattered spatially and over time. However, vital cell loss can create an erasure effect due to packetization and the effect is very objectionable. Considering the tight time constraint, retransmission is not feasible in packet video. It may also result in more severe congestion. Thus, error recovery has to be performed by the decoder alone. In this section, we will study the effect of errors on the output of the video coder, and propose some possible approaches to combat these effects.

### **6.7.1 Error Concealment**

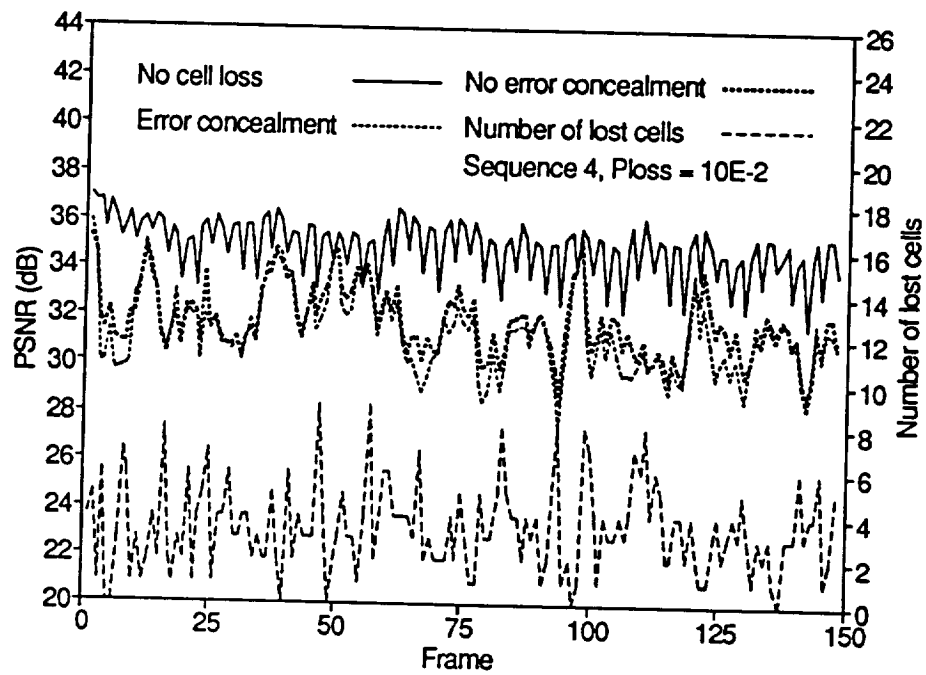
Error concealment can not provide perfect recovery, however, it can often provide a subjectively acceptable picture without increasing the transmission bandwidth. The concealment process is assisted by the transport format in detecting the image area which corresponds to the lost video data. Specific concealment procedures depend on the coding algorithm and on the level of complexity permissible in the decoder. For a layered coding scheme, there is auxiliary information available about the data in error. For example, in a subband coding scheme, if the cell containing information about pixels in one band is



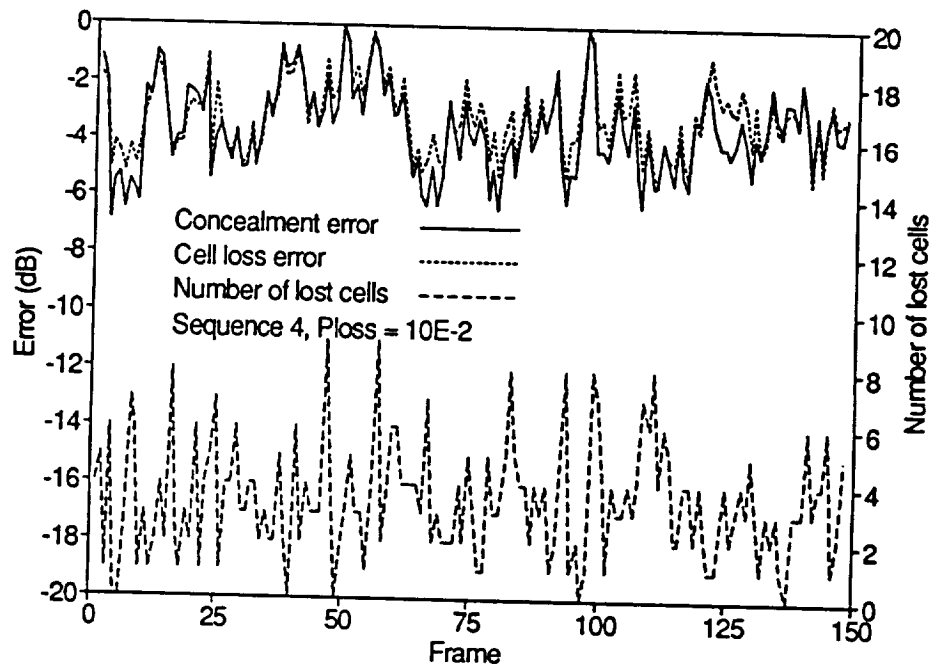
**Figure 6.6** Comparison of simulation results w/ and w/o concealment along with number of lost cells (Sequence 3).



**Figure 6.7** Concealment and cell loss error (Sequence 3).



**Figure 6.8** Comparison of simulation results w/ and w/o concealment along with number of lost cells (Sequence 4).



**Figure 6.9** Concealment and cell loss error (Sequence 4).

lost, information about these particular pixels contained in other bands can be used to make a reasonable prediction. In our simulator, spatial interpolation and motion compensated temporal replacement are used in combination to repair the damaged portions of the picture. Sequences 3 and 4 are used as test sequences because cell loss effect is very objectionable when cell loss occurs in I frames of the ADTV system. A simple interpolation which uses available data of top block in the same frame is adopted to conceal the errors for the block which lose its data. Since a cell loss generally causes loss of data in a series of blocks, the horizontal neighbors (left and right blocks) are not used for error concealment. Prediction using bottom block is also reasonable but it will introduce delay of one complete row of blocks in the decoder [74]. DC values and the five lowest order AC coefficients (in zig-zag order) are synthesized in order to reduce the blocking effect. Other AC coefficients are replaced with zeroes. Figure 6.6 shows three cases of no error, no concealment and with concealment for the *Susie* sequence at a cell loss rate of  $10^{-2}$ . This unrealistic high cell loss probability is chosen in order to clearly reveal the effect of cell loss. The number of lost cells for each frame is also indicated. For the case of no concealment, lost data is simply replaced with zero. Figure 6.7 shows an improvement of about 2 dB by using concealment. Figures 6.8 and 6.9 show the corresponding curves for the *Football* sequence. It is noticed that there is no improvement in PSNR by using this simple concealment scheme due to the low correlation between blocks in the *Football* sequence. However, from the recorded sequence, the *Football* sequence with concealment does perform better subjectively. On the other hand, since the *Susie* sequence is quite stationary, artifacts from loss of DC values are very visible and

the overall result is not acceptable.

### 6.7.2 Use of CLP Bit

In previous section, cell losses are equally distributed in the data without distinction. Now, the CLP bit in the cell header is set up following the principles described in the previous sections, then low priority cells are discarded first when the switching buffer overflows. Figure 6.10 shows the frame-by-frame PSNR performance of Sequence 7a when the transmission suffers cell loss with probability  $p_l = 10^{-2}$ . For the case of no priority scheme, cell loss is equally from four passes (not in headers and motion vectors) of the MBCPT coding scheme. In the priority scheme, only low priority cells (basically from pass 4) can get lost. Meanwhile, no correlation between cell losses is assumed although cell loss tends to occur in clusters. From the coded sequence, obvious impairment is observed for the case without the priority scheme. For the priority scheme, only slight artifacts are observed and overall performance is very pleasing. Figure 6.11 shows the improvement after adopting the priority scheme for Sequence 8a.

### 6.7.3 Partial Local Decoding (PLD)

If the network experiences a congestion with relative long duration, artifacts caused by cell loss are going to accumulate and propagate. Two possible approaches can be employed to counter this effect. First we can increase the frequency of intra-mode coding to avoid error propagation. However, increasing intra-mode coding may well increase the output rate and result in further network deterioration. The second approach to avoid error



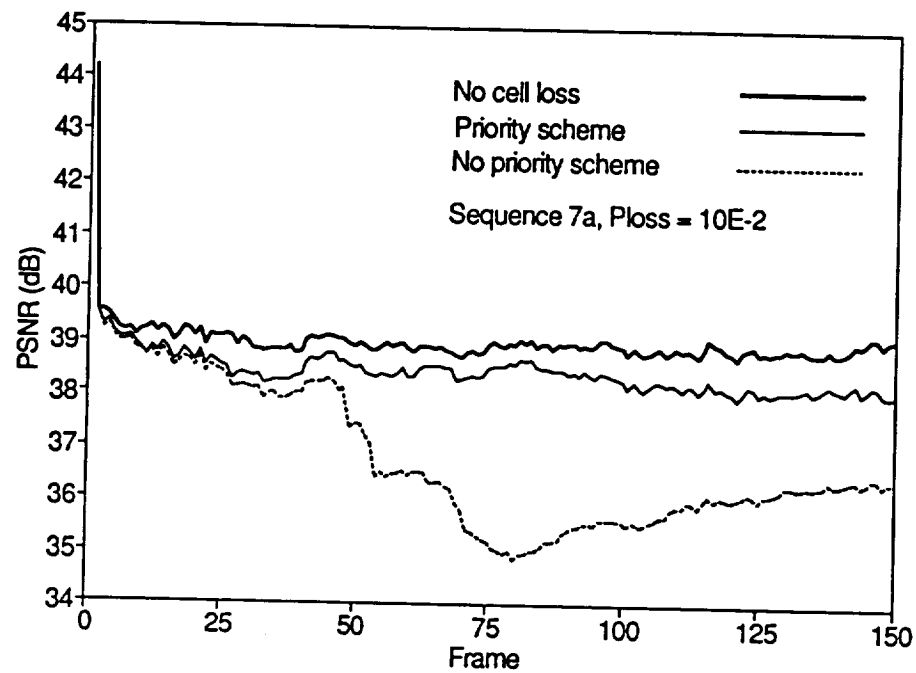


Figure 6.10 Performance of PSNR vs frame using priority scheme for Sequence 7a.

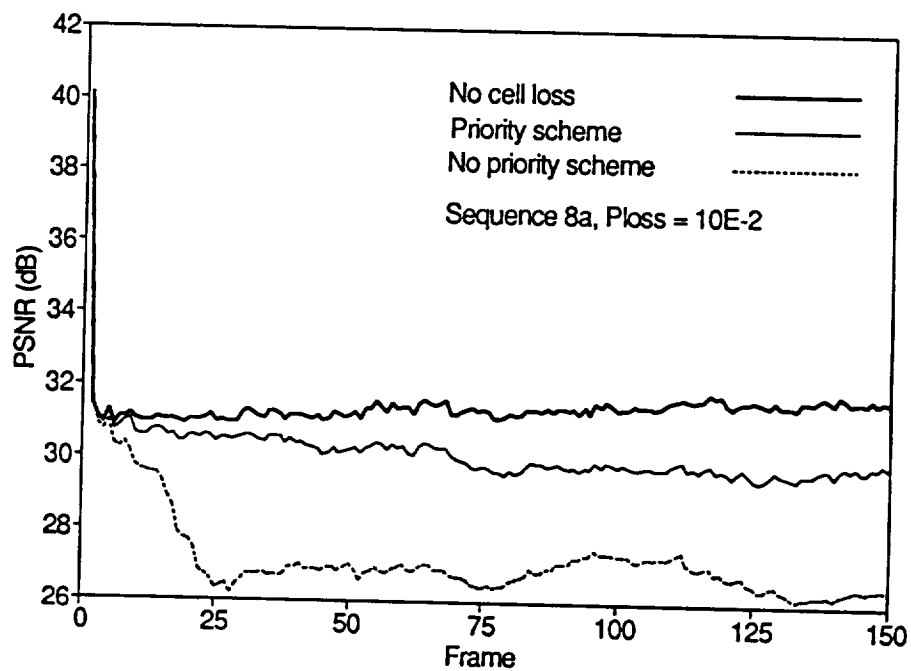
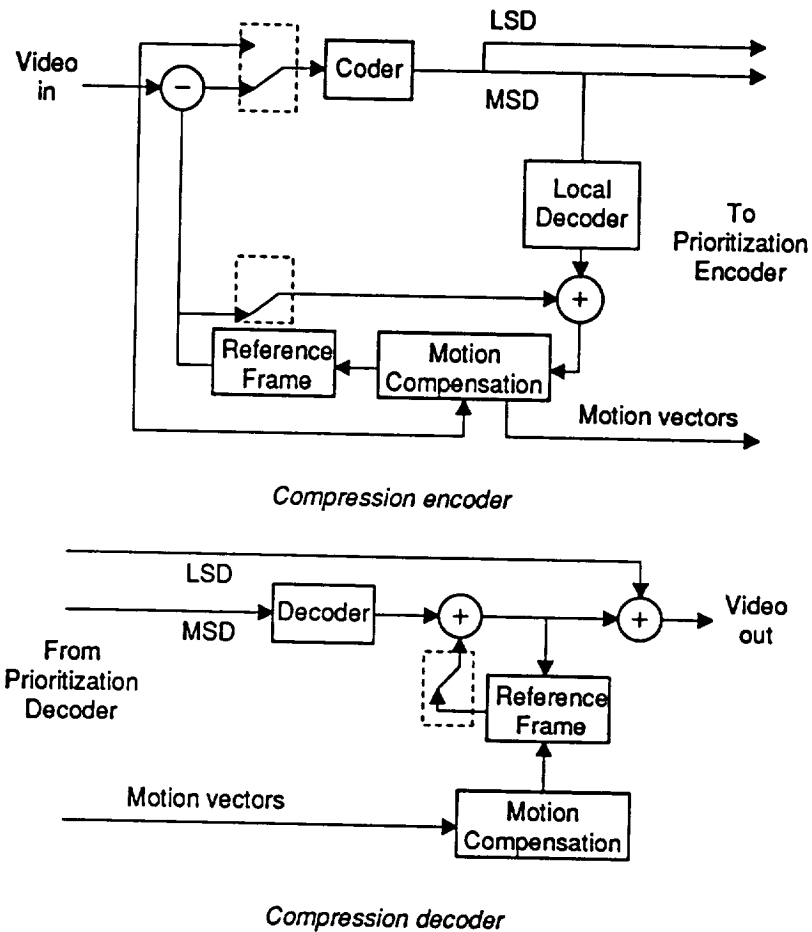


Figure 6.11 Performance of PSNR vs frame using priority scheme for Sequence 8a.



MSD: Most Significant Data  
 LSD: Least Significant Data

**Figure 6.12** A codec with partial local decoding.

propagation is a partial local decoding scheme [6] which is shown in Fig 6.12. In this scheme, only the most significant data (e.g. data of high priority cells) is used for local decoding. Therefore, the effect of losing low priority cells could only affect the current frame and will not propagate to the next frame since the lost least significant data is not used for reconstructing the reference frame. It is noted that partial local decoding is very easy to implement in the layered coding scheme (e.g. local decoding with some passes

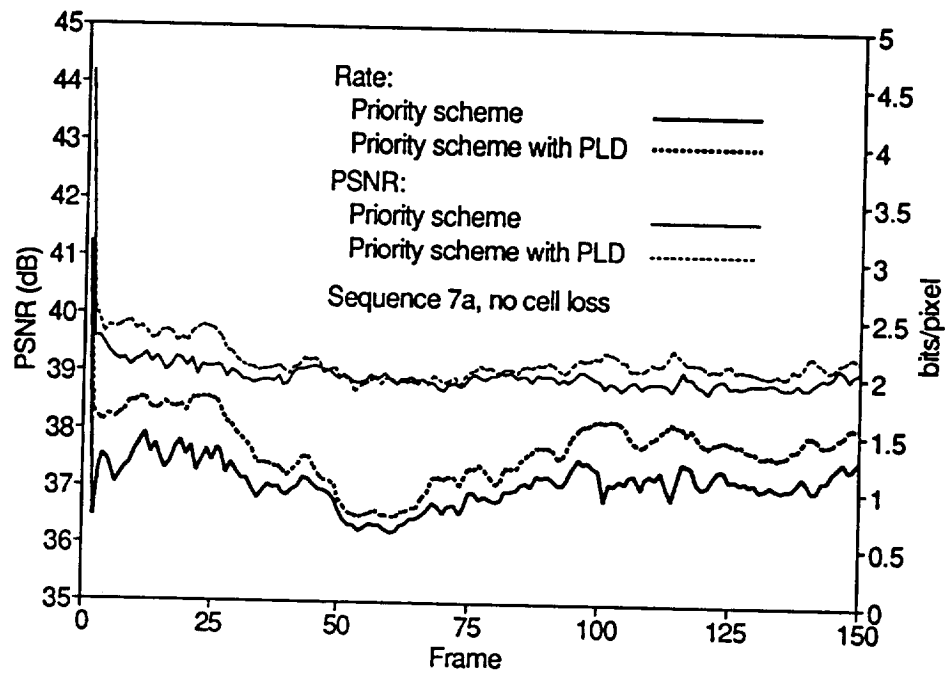


Figure 6.13 Performance of Sequence 7a using priority scheme w/ and w/o PLD.

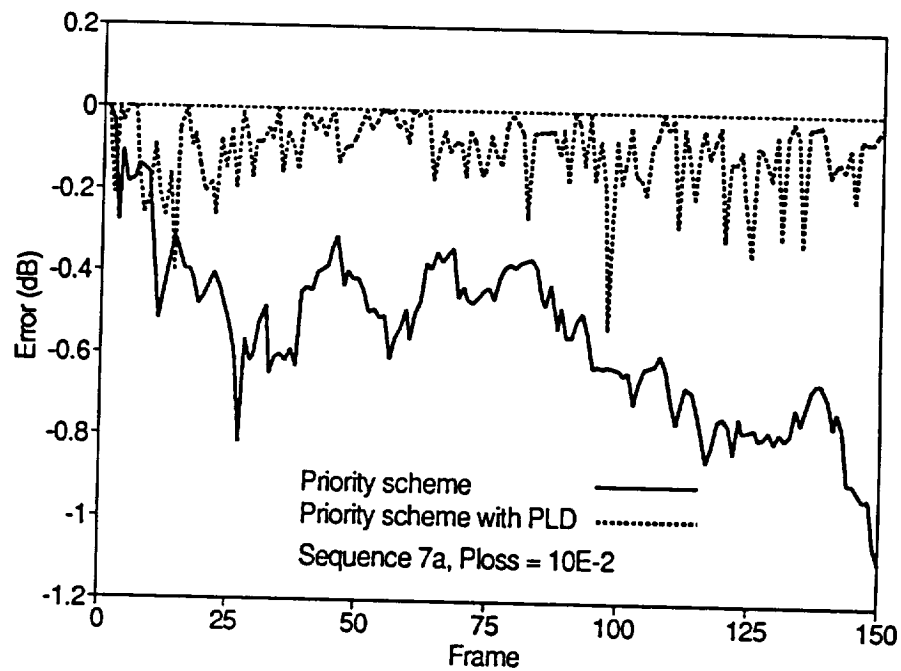


Figure 6.14 Improvement of Sequence 7a using PLD.

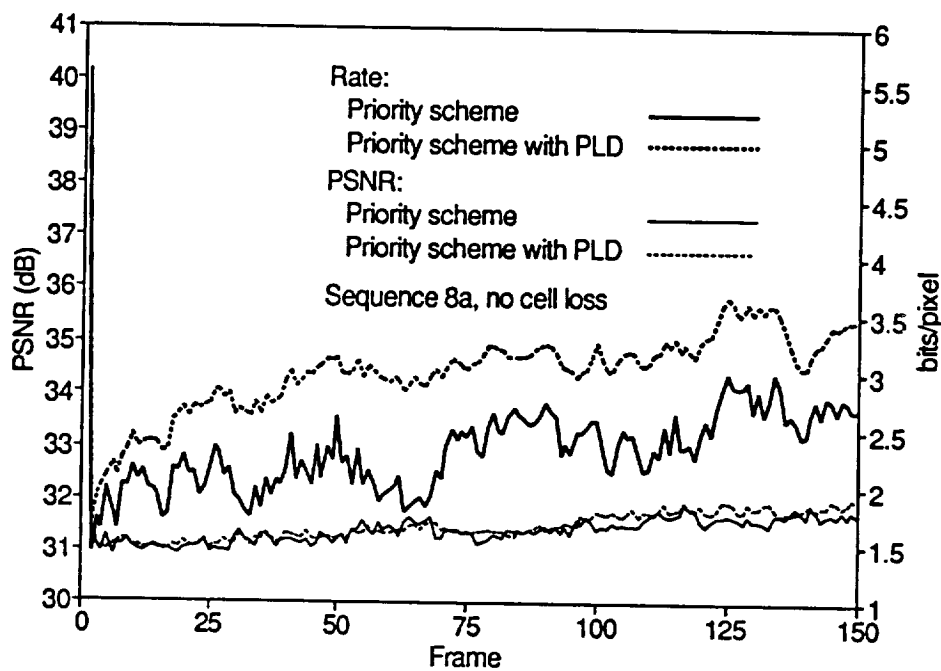


Figure 6.15 Performance of Sequence 8a using priority scheme w/ and w/o PLD.

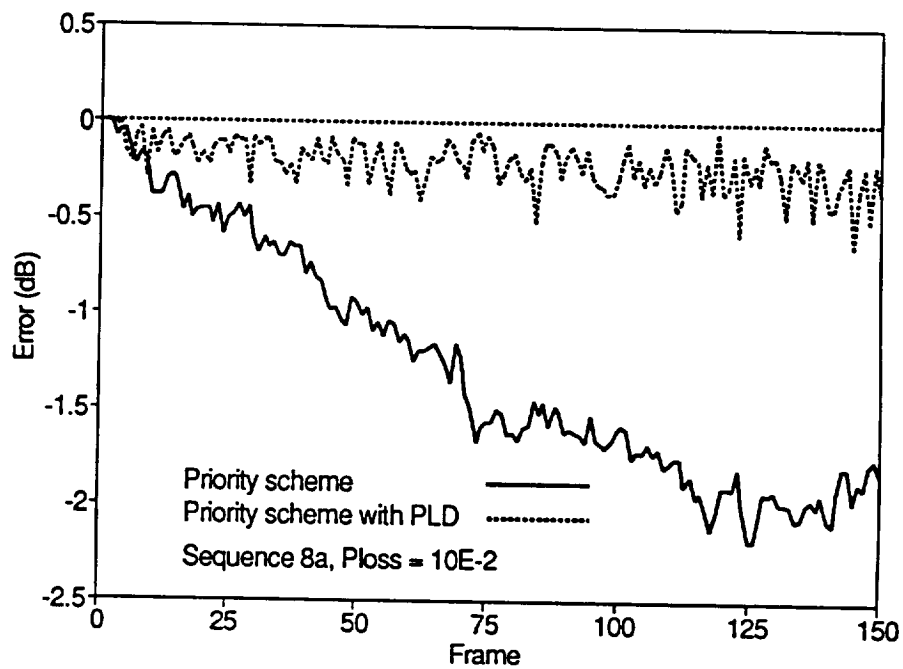
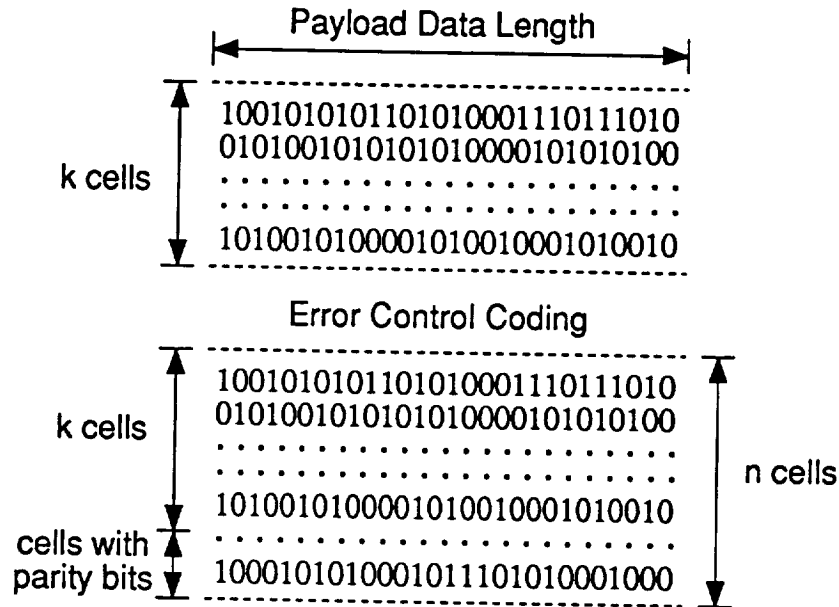


Figure 6.16 Improvement of Sequence 8a using PLD.

of MBCPT or some bands of subband coding). As for H.261 and ADTV, partial local decoding has to be integrated with the transport processor. That means the least significant data has to be packetized alone. Since only partial information from the current frame is used for prediction of the next frame, this results in a less efficient coding scheme. Simulation has been run using prioritized the MBCPT coding scheme. Only the data from the first three passes is defined as most significant data and used for local decoding. Figure 6.13 shows the coding performance of Sequence 7a using partial local decoding without cell loss. Although the output rate increases for about 25%, the ability of error protection improves significantly. Any low priority cell loss in network congestion results in a graceful degradation and will not affect the following frames. Figure 6.14 shows the improvement by using PLD when cell loss probability is again  $10^{-2}$ . It is noted that the cell loss error does not accumulate in this case. The result of subjective tests is very good and no artifact is observed. Figures 6.15 and 6.16 show the same results for the *Football* sequence.

#### 6.7.4 Other Possible Approaches

Vital information, like coding control data, is very important for video reconstruction. Error control coding, like CRC codes, can be applied in both directions along with and perpendicular to the packetization [5]. The former is for bit error in the data field like it is used in the cell header, while the latter is for cell loss. Figure 6.17 demonstrates the second case. The minimum distance that the error control coding should provide depends on the network's probability of cell loss, correlation of such loss and channel bit error



**Figure 6.17** Error control coding applied perpendicular to the direction of packetization.

rate.

It is also possible to decompose the image into several interleaved subimages at the encoder. Since strong correlation exists between these subimages, any cell loss from a particular subimage can be well recovered with linear interpolation using other subimages. Using the above concept, Sun and Manikopoulos [70] propose an adaptive interleaved coding scheme with vector quantization which is also suitable for ATM networks.

The Lapped Orthogonal Transform (LOT) has been proposed for video compression applications in ATM networks. LOT coders are designed to overlap blocks of coefficients before producing received images. It is claimed that this overlapping can greatly reduce the visible effect of any lost cells [75].

## 6.8 Some Notes

In this chapter we propose a complete set of guidelines for video codec design. No matter what kind of compression scheme is adopted, these principles can be applied in general. An efficient prioritized coding scheme is developed to generate red and green cells depending on cell's priority ranking. Green cells are generated with mean rate and red cells are generated with the difference of equivalent and mean bandwidth. Closely related to the design of the dual LB mechanism, the green cell can receive full protection and red cell will enjoy a good multiplexing environment after entering the network. The transport layer of the ADTV system has also been introduced in detail. Although the ADTV system is designed for RF channels, the function of packing data in an adaptation layer and video service layer can be applied in video transmission on ATM networks. Finally, some error control schemes were investigated and some pleasing results were obtained under the cell loss environment.

# Chapter 7

## Conclusions

Video communication is likely to be the most important service in the up-coming B-ISDN. Various applications, which include videophone, videoconferencing, HDTV broadcasting, multimedia, and personal communication network (PCN), have been explored in various detail. ATM-based B-ISDN provides great flexibility for service integration. The packet transport concept also allows image compression techniques to work to their full extent and variable bit rate coding becomes possible. Since the innovation of image compression techniques, it is almost possible now to transmit video with existing narrow-band channels. However, it is still exciting to have a new environment for the improvement of current video services and for the creation of new ones. Variable bit rate, constant quality coding is now the trend for video transmission since it can fully exploit correlation and redundancy in video sequences and provide a multiplexable output pattern to increase network utilization at the same time. Along with the advantages coming with ATM networks, new challenges have also emerged for coding specialists. Coding specialists have to design a new coding scheme which can take full



advantage of ATM networks. They also have to counter new error events like cell loss, cell delay, and delay jitter. The purpose of this dissertation is to provide new video coding principles which are based on the understanding and evaluation of ATM networks. We hope that the results and conclusions presented in this dissertation may contribute to create a guideline for the design of packet video codec.

Four video coding schemes have been introduced. The trade-off between quality and cost of each coding scheme has been compared. It has also been shown that variable bit rate, constant quality coding can be exercised using MBCPT. Characteristics of coded video sequences have been extensively explored and used as sample data in video source modelling. Several appropriate video source models for two representative video sources have been presented. With various goodness-of-fit tests, the validity of these models was justified. Goodness-of-fit tests include statistics tests and queueing behavior. An accurate video source model gives us the confidence of performing simulations in a network environment.

Network source management and congestion control are critical in designing video codec. We have proposed the dual leaky bucket algorithm for congestion control and been validated it as an effective policing tool for the network. Using this mechanism to monitor traffic flow not only gives high priority cells full protection, but also provides a good multiplexing environment for low priority cells. Meanwhile, by discarding violating cells from misbehaved connections or disconnecting the transmission directly, the dual leaky bucket algorithm can effectively discourage any source attempting a bad transmission. Other key control schemes, either preventive or reactive, have also been addressed in

detail. Priority schemes play a very important role in video transmission since, unlike data transmission, a significant part of video coding data is not essential in image reconstruction. Setting up the CLP bit efficiently can improve video quality as well as achieve an efficient transmission.

Based on the idea of the dual leaky bucket mechanism, a prioritized video coding scheme is implemented to realize efficient transmission by maintaining a good interaction with the network policing function. In this case, high/low priority cells will not be mistreated in the network. Simulations regarding cell loss effect have been performed. It is shown that a well-designed prioritized scheme can tolerate a high cell loss probability without visible artifacts. Partial local decoding provides an alternative when the network experiences a relatively long period of congestion. It effectively reduces the effects of error propagation. Several coding options have also been offered for the video coder to react to network congestion if an ECN bit is indicated.

There are still a lot of issues about B-ISDN left uncertain. It will require extensive efforts to reach the final agreement in order to clear the confusion among user, service provider, and equipment manufacturer and accelerate the pace of implementing B-ISDN. It is the process which evolves the coding concept along with the progress of B-ISDN we would like to devote our efforts to in the future.

# Bibliography

- [1] W. Verbiest, and L. Pinnoo, "A variable bit rate codec for asynchronous transfer mode networks," *IEEE J. Selected Areas Commun.*, vol. 7, no. 5, pp. 761-770, June 1989.
- [2] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE J. Selected Areas Commun.*, vol. 7, no. 5, pp. 771-781, June 1989.
- [3] J. C. Darragh, and R. L. Baker, "Fixed distortion subband coding of images for packet-switched networks," *IEEE J. Selected Areas Commun.*, vol. 7, no. 5, pp. 789-800, June 1989.
- [4] F. Kishino, K. Manabe, Y. Hayashi, and H. Yasuda, "Variable bit-rate coding of video signals for ATM networks," *IEEE J. Selected Areas Commun.*, vol. 7, no. 5, pp. 801-806, June 1989.
- [5] G. Karlsson, and M. Vetterli, "Packet video and its integration into the network architecture," *IEEE J. Selected Areas Commun.*, vol. 7, no. 5, pp. 739-751, June 1989.
- [6] Y.-C. Chen, K. Sayood, and D. J. Nelson, "A robust coding scheme for packet video," *IEEE Trans. Commun.*, vol. 40, no. 9, pp. 1491-1511, Sep. 1992.
- [7] S. E. Minzer, "Broadband ISDN and asynchronous transfer mode (ATM)," *IEEE Commun. Mag.*, vol. 27, no. 9, pp. 17-24, Sep. 1989.
- [8] CCITT Study Group XVIII, Draft Revision of Recommendation I.121, "Broadband aspects of ISDN," June 1990.
- [9] CCITT Study Group XVIII, Draft Recommendation I.150, "B-ISDN ATM functional characteristics," June 1990.

- [10] CCITT Study Group XVIII, Draft Recommendation I.211, "B-ISDN services aspects," June 1990.
- [11] S. Wolf, C. A. Dvorak, R. F. Kubichek, C. R. South, R. A. Schaphorst, and S. D. Voran, "How will we rate telecommunications system performance?" *IEEE Commun. Mag.*, vol. 29, no. 10, pp. 23-29, Oct. 1991.
- [12] B. Cole, "The technology framework," *IEEE Spectrum*, pp. 32-39, March 1993.
- [13] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall.
- [14] CCITT Study Group, Draft Revision of Recommendation CCITT H.261, "Video codec for audiovisual services at px64 kbits/s," March 1990.
- [15] The Advanced Television Research Consortium, *Advanced Digital Television: System Description*. Feb. 1991.
- [16] G. Karlsson, and M. Vetterli, "Subband coding of video for packet networks," *Optical Engineering*, vol. 27, no. 7, pp. 574-586, July 1988.
- [17] K. Sayood, Y. C. Chen, and X. Wang, "Low rate video coding," Semi-Annual Status Report to NASA Goddard Space Flight Center, Grant NAG 5-1612.
- [18] S-B. Ng, and L. Schiff, "Two-tier DPCM codec for videoconferencing," *IEEE Trans. Commun.*, vol. 37, no. 4, pp. 380-386, Apr. 1989.
- [19] K. Takahashi, and N. Ishii, "Robustness of data compression coding schemes for still pictures over noisy channels," *Proc. of ICC'90*, pp. 1035-1042, 1990.
- [20] P. Pancha, and M. E. Zarki, "A look at the MPEG video coding standard for variable bit rate video transmission," *INFOCOM'92*, pp. 85-94.
- [21] H. Heffes, and M. Lucantoni, "A markov modulated characterization of packeted voice and data traffic and related multiplexer performance," *IEEE J. Selected Areas Commun.*, vol. SAC-4, no. 6, pp. 856-868, Sep. 1986.
- [22] D. P. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical analysis and simulation study of video teleconference traffic in ATM Networks," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 2, no. 1, pp. 49-58, March 1992.
- [23] M. Nomura, T. Fujii, and N. Ohta, "Basic characteristics of variable bit rate video coding in ATM environment," *IEEE J. Selected Areas Commun.*, vol. 7, no. 5, pp. 752-760, June 1989.

- [24] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. Robbins, "Performance models of statistical multiplexing in packet video communications," *IEEE Trans. Commun.*, vol. COM-36, no. 7, pp. 834-843, July 1988.
- [25] P. Sen, B. Maglaris, N. Rikli, and D. Anastassiou, "Models for packet switching of variable bit-rate video sources," *IEEE J. Selected Areas Commun.*, vol. 7, no. 5, pp. 865-869, June 1989.
- [26] R. M. Rodriguez-Dagnino, M. R. K. Khansari, and A. Leon-Garcia, "Prediction of bit rate sequences of encoded video signals," *IEEE J. Selected Areas Commun.*, vol. 9, no. 3, pp. 305-314, Apr. 1991.
- [27] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257-286, Feb. 1989.
- [28] P. J. Brockwell, and R. A. Davis, *Time Series: Theory and Methods*. New York: Springer-Verlag.
- [29] L. Kleinrock, *Queueing Systems Volume I: Theory*. New York: Wiley-Interscience.
- [30] G. Ramamurthy, and B. Sengupta, "Modeling and analysis of a variable bit rate video multiplexer," *7th Specialist Seminar, International Teletraffic Congress*, Morristown, NJ, Oct. 1990.
- [31] R. Grünfelder, J. P. Cosmas, S. Manthorpe, and A. Odinma-Okafor, "Characterization of video codecs as autoregressive moving average processes and related queueing system performance," *IEEE J. Selected Areas Commun.*, vol. 9, no. 3, pp. 284-293, Apr. 1991.
- [32] H. Yamada, and S. Sumita, "A traffic measurement method and its application for cell loss probability estimation in ATM networks," *IEEE J. Selected Areas Commun.*, vol. 9, no. 3, pp. 315-324, Apr. 1991.
- [33] B. Melamed, D. Raychaudhuri, B. Senguta, and J. Zdepski, "TES-based traffic modeling for performance evaluation of integrated networks," *INFOCOM'92*, pp. 75-84.
- [34] A. M. Mood, F. A. Graybill, and D. C. Boes, *Introduction to the Theory of Statistics*, New York: McGraw-Hill.

- [35] J. S. Turner, "New directions in communications (or Which way to the information age?)," *IEEE Commun. Mag.*, vol. 24, no. 10, pp. 8-15, Oct. 1986.
- [36] M. Gerla, and L. Kleinrock, "Flow control: a comparative survey," *IEEE Trans. Commun.*, vol. COM-28, no. 4, pp. 553-574, Apr. 1980.
- [37] S. Yazid, and H. T. Mouftah, "Congestion control methods for BISDN," *IEEE Commun. Mag.*, vol. 30, no. 7, pp. 42-47, July 1992.
- [38] A. E. Eckberg, B. T. Doshi, and R. Zoccolillo, "Controlling congestion in B-ISDN/ATM: issues and strategies," *IEEE Commun. Mag.*, vol. 29, no. 9, pp. 64-70, Sep. 1991.
- [39] H. Gilbert, O. Aboul-Magd, and V. Phung, "Developing a cohesive traffic management strategy for ATM networks," *IEEE Commun. Mag.*, vol. 29, no. 10, pp. 36-45, Oct. 1991.
- [40] I. W. Habib, and T. N. Saadawi, "Controlling flow and avoiding congestion in broadband networks," *IEEE Commun. Mag.*, vol. 29, no. 10, pp. 46-53, Oct. 1991.
- [41] G. Ramamurthy, and R. S. Dighe, "A multidimensional framework for congestion control in B-ISDN," *IEEE J. Selected Areas Commun.*, vol. 9, no. 9, pp. 1440-1451, Dec. 1991.
- [42] A. Gersht, and K. J. Lee, "A congestion control framework for ATM networks," *IEEE J. Selected Areas Commun.*, vol. 9, no. 7, pp. 1119-1130, Sep. 1991.
- [43] H. Schulzrinne, J. F. Kurose, and D. Towsley, "Congestion control for real-time traffic in high-speed networks," *INFOCOM'90*, pp. 543-550.
- [44] M. G. Hluchyj, and M. J. Karol, "Queueing in high-performance packet switching," *IEEE J. Selected Areas Commun.*, vol. 6, no. 9, pp. 1587-1597, Dec. 1988.
- [45] T. Murase, H. Suzuki, S. Sato, and T. Takeuchi, "A call admission control scheme for ATM networks using a simple quality estimate," *IEEE J. Selected Areas Commun.*, vol. 9, no. 9, pp. 1461-1470, Dec. 1991.
- [46] R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application in high-speed networks," *IEEE J. Selected Areas Commun.*, vol. 9, no. 7, pp. 968-981, Sep. 1991.

- [47] H. Saito, and K. Shiimoto, "Dynamic call admission control in ATM networks," *IEEE J. Selected Areas Commun.*, vol. 9, no. 7, pp. 982-989, Sep. 1991.
- [48] C. Rasmussen, J. H. Sørensen, K. S. Kvols, and S. B. Jacobsen, "Source - independent call acceptance procedures in ATM networks," *IEEE J. Selected Areas Commun.*, vol. 9, no. 3, pp. 351-358, Apr. 1991.
- [49] J. Y. Hui, "Resource allocation for broadband networks," *IEEE J. Selected Areas Commun.*, vol. 6, no. 9, pp. 1598-1608, Dec. 1988.
- [50] G. Ramamurthy, and R. S. Dighe, "Distributed source control: a network access control for integrated broadband packet networks," *IEEE J. Selected Areas Commun.*, vol. 9, no. 7, pp. 990-1002, Sep. 1991.
- [51] M. Decina, and T. Toniatti, "On bandwidth allocation to bursty virtual connections in ATM networks," *Proc. of ICC'90*, pp. 844-851, Apr. 1990.
- [52] M. Decina, L. Faglia, and T. Toniatti, "Bandwidth allocation and selective discarding for variable bit rate and bursty data calls in ATM networks," *Proc. INFOCOM'91*, pp. 1386-1393.
- [53] T. Kamitake, and T. Suda, "Evaluation of an admission control scheme for an ATM network considering fluctuations in cell loss rate," *IEEE Globecom'89*, pp. 1774-1780, Nov. 1989.
- [54] B. Kraimeche, and M. Schwartz, "Analysis of traffic access control strategies in integrated service networks," *IEEE Trans. Commun.*, vol. Com-33, no. 10, pp. 1085-1093, Oct. 1985.
- [55] S-Q. Li, "Overload control in a finite message storage buffer," *IEEE Trans. Commun.*, vol. 37, no. 12, pp. 1330-1338, Dec. 1989.
- [56] I. Cidon, I. Gopal, and R. Guérin, "Bandwidth Management and Congestion Control in plaNET," *IEEE Commun. Mag.*, vol. 29, no. 10, pp. 54-64, Oct. 1991.
- [57] CCITT Study Group XVIII, Draft Recommendation I.311, "B-ISDN general network aspects," June 1990.
- [58] E. P. Rathgeb, "Modelling and performance comparison of policing mechanisms for ATM networks," *IEEE J. Selected Areas Commun.*, vol. 9, no. 3, pp. 325-334, Apr. 1991.
- [59] G. Gallassi, G. Rigolio, and L. Fratta, "ATM: Bandwidth assignment and bandwidth enforcement policies," *IEEE Globecom 89*, pp. 1788-1793, Nov. 1989.

- [60] P. Tran-Gia, and H. Ahmadi, "Analysis of a discrete-time  $G^{(X)}/D/1-S$  queueing system with applications in packet-switching systems," *INFOCOM'88*, pp. 861-870.
- [61] S. J. Golestani, "Congestion-free transmission of real-time traffic in packet networks," *INFOCOM'90*, pp. 527-536.
- [62] D. W. Petr, and V. S. Frost, "Optimal packet discarding: An ATM-oriented analysis model and initial results," *INFOCOM'90*, pp. 537-542.
- [63] H. Kröner, G. Hébuterne, P. Boyer, and A. Gravey, "Priority Management in ATM switching nodes," *IEEE J. Selected Areas Commun.*, vol. 9, no. 3, pp. 418-427, Apr. 1991.
- [64] J. S-C. Chen, and R. Guérin, "Performance study of an input queueing packet switch with two priority classes," *IEEE Trans. Commun.*, vol. 39, no. 1, pp. 117-126, Jan. 1991.
- [65] N. Yin, and M. G. Hluchyj, "A dynamic rate control mechanism for source coded traffic in a fast packet network," *IEEE J. Selected Areas Commun.*, vol. 9, no. 7, pp. 1003-1012, Sep. 1991.
- [66] M. Sidi, W-Z. Liu, I. Cidon, and I. Gopal, "Congestion control through input rate regulation," *IEEE Globecom 89*, pp. 1764-1768, Nov. 1989.
- [67] K. Bala, I. Cidon, and K. Sohraby, "Congestion control for high speed packet switched networks," *INFOCOM'90*, pp. 520-526.
- [68] R. Guérin, and L. Gǔn, "A unified approach to bandwidth allocation and access control in fast packet-switched networks," *INFOCOM'92*, pp. 1-12.
- [69] D-S. Lee, K-H. Tzou, and S-Q. Li, "Control and analysis of video packet loss in ATM networks," *Optical Engineering*, vol. 30, no. 7, pp. 955-964, July 1991.
- [70] H. Sun, and C. N. Mainkopoulos, "Adaptive interleaved vector quantization for image transmission," *Proc. of ICC'89*, pp. 1355-1360, 1989.
- [71] T. Aoyama, I. Tokizawa, and K. Sato, "ATM VP-based broadband networks for multimedia services," *IEEE Commun. Mag.*, vol. 31, no. 4, pp. 30-39, Apr. 1993.
- [72] P. Skelly, S. Dixit, and M. Schwartz, "A histogram-based model for video traffic behavior in an ATM network node with an application to congestion control," *INFOCOM'92*, pp. 95-104, Florence Italy, 1992.



- [73] K. Joseph, S. Ng, R. Siracusa, D. Raychaudhuri, and J. Zdepski, "Prioritization and transport in the ADTV digital simulcast system," *IEEE Trans. Consumer Electronics*, vol. 38, no. 3, pp. 319-323, Aug. 1992.
- [74] H. Sun, K. Challapali, and J. Zdepski, "Error concealment in digital simulcast AD-HDTV decoder," *IEEE Trans. Consumer Electronics*, vol. 38, no. 3, pp. 108-117, Aug. 1992.
- [75] P. Haskell, and D. Messerschmitt, "Reconstructing lost video data in a lapped orthogonal transform based coder," *ICASSP*, vol. 4, pp. 1985-1988, 1990.